



## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **06022152 A**

(43) Date of publication of application: 28.01.94

(51) Int. Cl

H04N 1/415

**G06F 15/66**

H03M 7/30

H04N 1/41

H04N 7/133

(21) Application number: 04173443

(71) Applicant: **CANON INC**

(22) Date of filing: 30.06.92

(72) Inventor: MIYAKE NOBUTAKA  
NAKAYAMA TADAYOSHI  
HASEGAWA KAZUhide  
OKI JOJI

(54) PICTURE PROCESSING UNIT

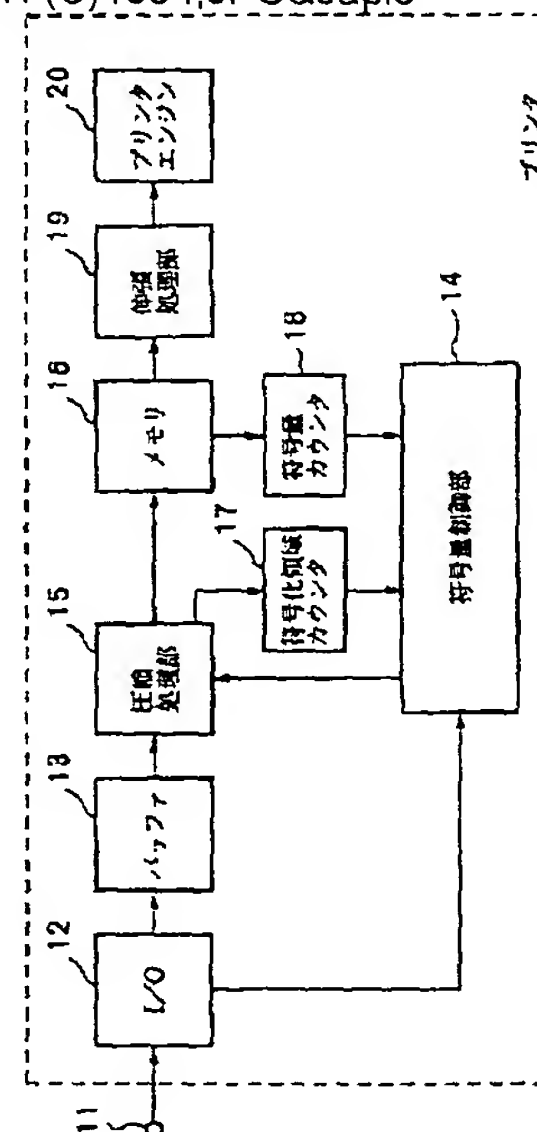
(57) Abstract:

**PURPOSE:** To reduce the transfer time of picture information by controlling a variable length code quantity by one picture coding to a memory capacity of the processing unit or below.

**CONSTITUTION:** A code quantity control section 14 receives picture size information outputted from an I/O 12 to a printer. The control section 14 calculates a reference code quantity per block based on picture size information. The picture information is stored in a buffer 13 by one line block from the I/O 12 and sent to a compression processing section 15. The processing section 15 applies coding processing according to the information from the control section 14 and stores the code by one block in a memory 16. A coding area counter 17 counts the number of blocks subjected to coding processing and obtains a permissible area capacity for a prescribed compression condition based on the count and the reference code quantity. On the other hand, the code quantity stored in the memory 16 is counted by a code quantity counter 18 and a succeeding compression condition is revised based on the comparison between the count and the permissible area capacity at the control

section 14. Thus, the code quantity is controlled by one path and the transfer time is reduced.

COPYRIGHT: (C)1994,JPO&Japio



(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-22152

(43)公開日 平成6年(1994)1月28日

(51)Int.Cl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
H 0 4 N 1/415		9070-5C		
G 0 6 F 15/66	3 3 0 H	8420-5L		
H 0 3 M 7/30		8522-5J		
H 0 4 N 1/41		B 9070-5C		
7/133	Z			

審査請求 未請求 請求項の数17(全 32 頁)

(21)出願番号 特願平4-173443

(22)出願日 平成4年(1992)6月30日

(71)出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 三宅 信孝

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

(72)発明者 中山 忠義

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

(72)発明者 長谷川 一英

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

(74)代理人 弁理士 大塚 康徳 (外1名)

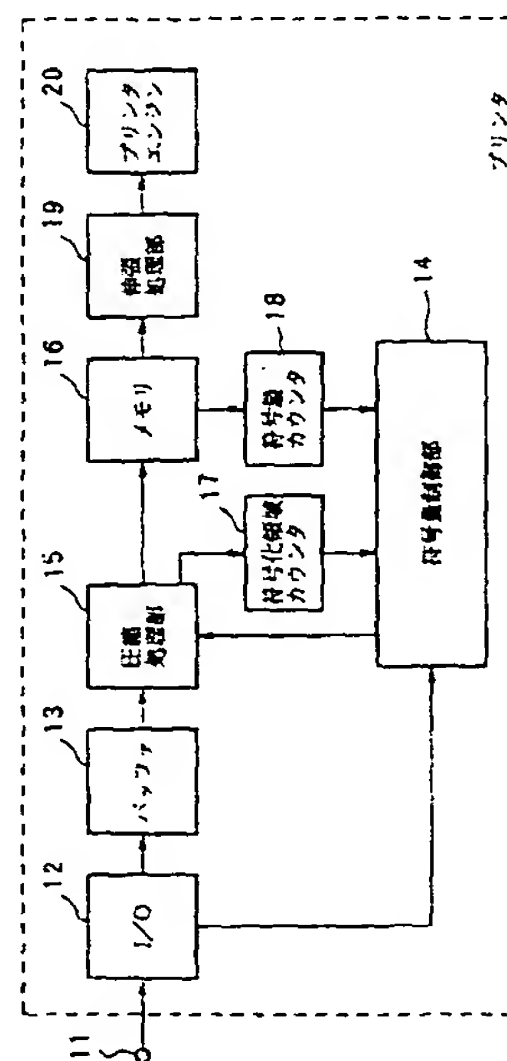
最終頁に続く

(54)【発明の名称】 画像処理装置

#### (57)【要約】

【目的】 ホストコンピュータからの画像情報の転送時間が少なくて済み、また画像情報の偏りに合った良好な圧縮処理を実現し、少量の所有メモリで製品化を実現できる。

【構成】 画像情報を格納するメモリを備え、累積符号量と累積符号化領域の情報により、逐次、符号量を制御することにより、1回のパスで画像の変長符号を一定のメモリ内に符号量を制御することができる。



## 【特許請求の範囲】

【請求項 1】 最大出力面積に値する記憶容量よりも少量の記憶容量を有する記憶手段を備える画像処理装置であつて、  
画像情報をブロック化して可変長符号化する符号化手段と、

該符号化手段による 1 回の画像符号化により可変長符号量を前記記憶手段の容量以下に制御する制御手段とを備えることを特徴とする画像処理装置。

【請求項 2】 前記制御手段は、累積符号化領域の情報と累積符号量の情報により前記可変長符号量を制御することを特徴とする請求項 1 記載の画像処理装置。

【請求項 3】 前記制御手段は、直交変換係数の量子化条件を変化させることにより前記可変長符号量を制御することを特徴とする請求項 1 記載の画像処理装置。

【請求項 4】 前記制御手段は、K ブロックの累積符号量と K ブロックでの設定符号量との比較により前記可変長符号量を制御することを特徴とする請求項 1 記載の画像処理装置。

【請求項 5】 入力画像信号を複数画素からなるブロックに分割するブロック化手段と、  
該ブロック化手段で分割したブロックごとに系列変換する変換手段と、

該変換手段で変換された変換係数を量子化マトリクスを用いて量子化する量子化手段と、  
該量子化手段で量子化された変換係数を可変長符号を用いて符号化する符号化手段と、

該符号化手段で符号化された符号量を累積する符号量累積手段と、

該符号量累積手段での累積符号量を元に前記量子化手段で用いる量子化マトリクスを切り換える量子化マトリクス設定手段と、

前記符号量累積手段において累積された符号量から増加率を判定する増加率判定手段と、

該増加率判定手段での判定結果から前記量子化マトリクス設定手段での量子化マトリクスの切り換えに用いる閾値を切り換える閾値切り換え手段とを備えることを特徴とする画像処理装置。

【請求項 6】 符号化データを格納する記憶部の容量に対する発生符号量の割合を算出する第 1 の算出手段と、  
全画像データに対する符号化済み画像データ量り割合を求める第 2 の算出手段と、

該第 1 及び第 2 の算出手段から得られる各々の数値に基づいて圧縮率の制御を行なう制御手段とにより画像符号化を行うことを特徴とする画像処理装置。

【請求項 7】 更に画像データを量子化処理する量子化手段を備え、前記制御手段による圧縮率の制御は前記量子化手段の量子化条件の変更により行なうことを特徴とする請求項 6 記載の画像処理装置。

【請求項 8】 前記符号化済み画像データ量の割合は、

符号化済み画像データ量を係数した係数結果を全画像データ量で除算することによって得ることを特徴とする請求項 6 記載の画像処理装置。

【請求項 9】 全画像データ量に対する画像の所定データ量の割合を求める割合算出手段と、

符号化した画像データ量に応じて累積加算を行なう累積加算手段とを備え、

第 2 の算出手段は少なくとも、該累積加算手段及び割合算出手段を用いて前記符号化済み画像データ量の割合を求めることを特徴とする請求項 6 記載の画像処理装置。

【請求項 10】 符号化処理により発生した符号量を計数する符号量計数手段と、

画像データ量に比例する基準符号量なる信号を生成する基準符号量生成手段と、

該符号量計数手段と該基準符号量生成手段各々から出力される値に基づいて圧縮率の制御を行なう制御手段とを備えることを特徴とする画像処理装置。

【請求項 11】 更に符号化処理に量子化手段を有し、制御手段による圧縮率の制御は前記量子化手段の量子化条件の変更により行なうことを特徴とする請求項 10 記載の画像処理装置。

【請求項 12】 基準符号化量生成手段は、符号化データを格納する記憶部の容量と符号化する画像の全データ量や属性を基に、所定の画像データ量に対する割り当て符号量を演算する演算手段と、該演算手段で演算した割り当て符号量を累積する累積手段とを含むことを特徴とする請求項 10 記載の画像処理装置。

【請求項 13】 全画像情報が複数の小画像より成る場合、符号量計数手段は各小画像ごとに符号量を累積し、基準符号量生成手段は各小画像ごとに割り当て符号量を累積することを特徴とする請求項 12 記載の画像処理装置。

【請求項 14】 符号化量計数手段は、画像の全符号量を累積して計数することを特徴とする請求項 10 記載の画像処理装置。

【請求項 15】 符号量計数手段は、所定の画像データ量ごとに符号量を累積することを特徴とする請求項 10 記載の画像処理装置。

【請求項 16】 制御手段は、符号量計数手段の出力値が該基準符号量生成手段の出力値より小さくなるまで所定の画像データを圧縮率を変えながら繰り返し符号化させることを特徴とする請求項 15 記載の画像処理装置。

【請求項 17】 符号量計数手段は 符号化データを格納する記憶部のアドレス発生手段を含むことを特徴とする請求項 10 記載の画像処理装置。

## 【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は画像処理装置に関し、例えば、画像情報を格納するメモリを所有する画像出力装置である画像処理装置、又は、入力画像をエントロピー

符号化する画像処理装置に関するものである。

【0002】

【従来の技術】近年は画像表示部の高解像度化とフルカラー化、並びに半導体技術の著しい発展に伴ない、高解像のフルカラー情報をメモリに格納し、それを画像表示部に出力する装置が技術的に可能になってきた。とはいっても、大容量の画像データを格納するには、非常に多くのメモリを必要とし、これが装置のコストアップの大きな要因になっていた。例えば、A3の用紙サイズで400ドット・パー・インチ（以下dpiと略す）の解像度、レッド（赤）、ブルー（青）、グリーン（緑）の各色が、8ビットのフルカラー画像の場合、およそ96メガバイトの情報量になり、この情報を全て格納するには4メガビットメモリを192ケも必要としていた。

【0003】そこで、画像圧縮という手法を用いて、画像の情報量を減らし、少ないメモリでフルカラー画像を格納出来るような工夫が従来なされてきた。圧縮の方法には、いろいろなものがあるが、変換符号化として、現在よく知られている離散コサイン変換（以下、DCT変換と略す）を例にとって従来の圧縮方法の説明を図21を用いて以下に行なう。

【0004】同図において、401は画像データを入力する入力端子、402は画像データの一部を一時的に格納するためのバッファ、403は画像をDCT変換するDCT変換部、404はDCT変換された後の各周波数成分を再量子化するための量子化部、405はビット単位のデータをバイト単位もしくはワード単位のデータにまとめるためのパッキング処理部、406は圧縮符号化されたデータを出力する出力端子、407は画像の横サイズを入力する入力端子、408は画像の縦サイズを入力する入力端子、409は画像の横サイズと縦サイズからDCT変換する際の基本単位であるブロックに対して、何ビットの情報が割り当て可能かを演算する符号量演算部、410は1ブロック当たりの符号量からDCT変換後の各周波数成分に対して何ビットの情報を割り当てるかを定めるビット配分演算部である。

【0005】画像出力装置としては、図20に示す圧縮処理部以外にホストコンピュータ等から送られるデータの転送を制御するインターフェース部、圧縮したデータを格納するための記憶部、該記憶部から読み出した圧縮データを伸張して画像データに復元する伸張部、復元された画像データを出力する画像出力部等がある（図22参照）。

【0006】コンピュータ等の外部装置から送られてきたデータは501のインターフェース部にて、画像のヘッダ情報と実際の画像情報とに分けられ、ヘッダ情報にある画像の横サイズ情報は、図20における端子407に縦サイズ情報は端子408に入力されると共に、画像情報は端子401に入力される。端子407、408に入力された画像の横サイズ、縦サイズ情報は、符号量演

算部409に送られ、1ブロックに割り当て可能なビット数が演算され出力される。

【0007】これは、あらかじめ定められた容量を持つ記憶部503を最大限有効に活用するために行なう処理である。例えば、記憶部503の全メモリ容量が128メガビット、画像の横サイズが4000、縦サイズが6000、1画素が赤（R）、青（B）、緑（G）各色8ビットから成り、DCTの変換ブロックの大きさが8×8画素の場合、まず、全ブロック数は4000×6000÷（8×8）×3=1125000となる。

【0008】これより1ブロック当たりのビット数は128×2<sup>20</sup>÷1125000=119となる。この値は、ビット配分演算部410に送られ、トータルで119ビットとなるように、DCT変換後の各周波数成分に対して1つ1つ符号化ビット数を割り当てる。この割り当ての一例を図22に示す。一方、端子401に入力された画像情報は、バッファ402にて、一次的に格納される。通常、画像情報はラスタ状に運転されるが、この転送順序では、DCT変換部403にてDCT変換を行なうことが出来ない。そこで、バッファ402で8ライン以上のデータを格納した後、8×8画素のブロック単位でデータを読み出し、DCT変換部403に画像情報データを送る。DCT変換部403は、ブロック単位で該画像情報データをDCT変換し、変換後の周波数成分を量子化部404へ送る。

【0009】量子化部404は、該周波数成分データと各周波数成分に対応したビット配分情報を受けとり、ビット配分に応じた量子化を行なう。ここでの量子化は、一般に非線型量子化を用いる。量子化された後のビット情報は、パッキング処理部405へ送られ、バイト単位、もしくはワード単位の情報にパッキングされて、圧縮データとして端子406へ出力される。該圧縮データは、図21において、圧縮処理部502から出力され、記憶部503に格納される。

【0010】更に、従来この種の画像処理装置におけるカラー静止画符号化の国際標準化方式として、JPEG (Joint Photographic Experts Group)にて提案されているベースラインシステム（基本方式）の符号化方式がある。この方式の詳細説明は省略するが、概略はDCT後の量子化係数をハフマン符号化により、可変長符号化する方式である。以上の方式は、可変長符号化の為、どのような画像でも、ある一定のメモリ内に制御するのは困難である。

【0011】また、各ブロックでDCT後の量子化係数を固定長符号化する方式では、画像の情報量の偏りにより、あるブロックでは画質が劣化するということが発生してしまう。そこで、近年、電子的に撮像してフロッピーディスクやICカード等に格納する、いわゆる電子撮像カメラに応用する為に、上述のJPEGの方法を利用した符号量の制御方法が提案されている。



【0012】図19に、前記電子撮像カメラの応用で提案されている技術を示す。図19において、1000はフレームメモリを示し、撮像部（不図示）から入力した画像データが格納される。その後、ブロック化回路（不図示）にて8×8画素サイズごとにブロック化されて、DCT処理1001がなされる。DCT後の変換係数は、量子化器1002にて量子化処理される。この場合、パスカウンタ1003は1パス目ということにより、スイッチ1004は端子Aに接続してあり、あらかじめ設定してある量子化テーブル1005の値が量子化器1002に送信される。

【0013】量子化後の係数は、ジグザグスキャン回路1006により、低周波域から高周波域に一次的に並び換えられる。量子化係数は、“0”のラン長と、

“0”以外の有意係数に分類され（不図示）、ハフマン符号器1007において、ハフマン符号化される。パスカウンタ1003は、スイッチ1008では端子Dに接続されていて、符号量測定器1009にて符号量の測定が行われる。そうして、フレームメモリ1000に格納されている画像をまず、全画像分符号化してみ、符号量測定器1009は一画面分の符号量を測定する。

【0014】この測定結果に基づいて、スケーリングファクター変更部1010では、ある一定の符号量に納まる様に、量子化テーブルに積算する値を経験則から決定し、掛け算器1011に送信する。2パス目は、パスカウンタはスイッチ1004ではB端子に、スイッチ1008ではC端子に接続されている。再び、フレームメモリ1000から画像情報を入力してDCT処理を行い、今度は、スケーリングファクターを乗じた量子化テーブルの情報により量子化処理を施し、ジグザグスキャン回路、ハフマン符号器を通して、端子Cにより符号データが出力できる。

【0015】この符号データは、1画面分の情報でみると、ある一定内の符号量に制御されている。

【0016】

【0017】

【発明が解決しようとする課題】しかし、一般的に画像データというのは情報の偏りが大きい。例えば、雲1つ無い青空の領域は、直流成分のみでほとんど表現できてしまうのに対して、細かい木の枝が密集した領域は、直

【0018】こういったことから、画像情報を保存するために必要な符号量（ビット数）はブロックごとに大きく異なり、その差は十倍以上になる。

【0019】しかしながら、図20に示した上記従来例では、各ブロックに同一の符号量を割り当てているため、情報量の少ないブロックではメモリが有効に活用されないという問題点があり、実際に有効に使われるメモリの数倍の容量を用意する必要があった。更に、上述した図19の方式においては、ある一定のメモリを有する

プリンタに応用した場合、入力する画像サイズが異なる場合があるという欠点があり、また、フレームメモリを有していない為に複数パスは困難である欠点（接続先のホストコンピュータから複数回送信してもらうと、転送時間が増大する）があった。

【0020】

【課題を解決するための手段】本発明は上述の課題を解決することを目的としてなされたもので、上述の課題を解決する一手段として以下の構成を備える。即ち、最大出力面積に値する記憶容量よりも少量の記憶容量を有する記憶手段を備える画像処理装置であつて、画像情報をブロック化して可変長符号化する符号化手段と、該符号化手段による1回の画像符号化により可変長符号量を前記記憶手段の容量以下に制御する制御手段とを備える。

【0021】また、入力画像信号を複数画素からなるブロックに分割するブロック化手段と、該ブロック化手段で分割したブロックごとに系列変換する変換手段と、該変換手段で変換された変換係数を量子化マトリクスを用いて量子化する量子化手段と、該量子化手段で量子化された変換係数を可変長符号を用いて符号化する符号化手段と、該符号化手段で符号化された符号量を累積する符号量累積手段と、該符号量累積手段での累積符号量を元に前記量子化手段で用いる量子化マトリクスを切り換える量子化マトリクス設定手段と、前記符号量累積手段において累積された符号量から増加率を判定する増加率判定手段と、該増加率判定手段での判定結果から前記量子化マトリクス設定手段での量子化マトリクスの切り換えに用いる閾値を切り換える閾値切り換え手段とを備える。

【0022】更にまた、符号化データを格納する記憶部の容量に対する発生符号量の割合を算出する第1の算出手段と、全画像データに対する符号化済み画像データ量り割合を求める第2の算出手段と、該第1及び第2の算出手段から得られる各々の数値に基づいて圧縮率の制御を行なう制御手段とを備え、画像符号化を行う。また、符号化処理により発生した符号量を計数する符号量計数手段と、画像データ量に比例する基準符号量なる信号を生成する基準符号量生成手段と、該符号量計数手段と該基準符号量生成手段各々から出力される値に基づいて圧縮率の制御を行なう制御手段とを備える。

【0023】

【作用】以上の構成において、ホストコンピュータからの画像情報の転送時間が少なく済み、また可変長符号化を用いているため、画像情報の偏りに合った良好な圧縮処理が実現できる。その結果、大容量を必要とするプリンタ、特にカラープリンタにおいても、少量の所有メモリで製品化を実現できる。

【0024】また、量子化マトリクスの切り換えに用いる閾値を画像の位置における符号化情報量に応じて適応的に切り換えることにより、画像全体での量子化幅を段

10

20

30

40

50

階的に変化させることができる。また、符号後の符号量  
を目標の値に固定することもできる。更にまた、第1及  
び第2の算出手段から得られる各々の数値に基づいて圧  
縮率の制御を行なうことにより、該記憶部を有効に使用  
し、該記憶部を従来より少ない容量で実現できる。

【0025】

【実施例】以下に添付図面を参照して、本発明に係る一  
実施例を詳細に説明する。

＜第1の実施例＞図1は本発明の画像処理装置をプリン  
タに適用した第1の実施例の構成を示すブロック図であ  
る。

【0026】図1において、11は入力端子であり、本  
実施例のプリンタと接続している情報の入力源を示して  
いる。情報を出力する接続機器としては、ホストコンピ  
ュータ、また近年、実現化しているページ記述言語のイン  
タプリタを搭載しているフォーマッタ等が挙げられ  
る。12はプリンタのI/Oを示し、前述した接続機器  
との情報のやりとりを司る。13はバッファメモリを示  
し、数ライン分、例えば、ブロック化を施して直交変換  
を利用する圧縮方式の場合には、最低でもそのブロック  
ライン分の情報を格納するメモリである。14は符号量  
制御部を示し、I/Oから画像サイズの情報を受信し  
て、目標の圧縮率を設定して、符号量が一定値以内な納  
まる様に制御するブロックを示す。

【0027】15は圧縮処理部を示し、バッファ13に  
格納してある画像情報を入力して、画像の冗長度を減ら  
して可変長符号化するブロックである。16はメモリを  
示し、圧縮処理部14からの符号を格納するブロックで  
ある。当然、このメモリは本実施例に係るプリンタの最  
大出力サイズ分のメモリ容量よりも少量で構成されてい  
るメモリである。

【0028】17は符号化領域カウンタを示し、圧縮処  
理部15において符号化を進めていく時に、例えば、ブ  
ロック化して符号化を処理する際に現在までに処理した  
ブロック数をカウントする。18は符号量カウンタを示  
し、現在までにメモリに格納された符号量をカウントす  
る。19は伸張処理部を示し、メモリ16から符号を取  
りだし、プリンタエンジン20の都合のよいように（エン  
ジンのプロセススピードのタイミングに合わせて）復  
号していくブロックである。破線で囲んである部分がプ  
リントに相当する。

【0029】図2は、図1の符号量制御部14の詳細構  
成の一例を示している。破線で囲んだ部分が符号量制御  
部14に相当する。また、図3は、図1の圧縮処理部1  
5の詳細構成の一例を示している。破線で囲んだ部分が  
圧縮処理部15に相当する。図1～図3を用いて本実施  
例の信号の流れを説明する。

【0030】まず、図1の符号量制御部14がI/O1  
2からプリンタに出力する画像サイズ情報を受信する。  
画像サイズは、画像領域が長方形の場合は、＜横画素数

＞、＜縦画素数＞という形式で接続したホストコンピ  
ュータから受信する方式、長方形以外の画像領域の場合  
は、＜画像領域の面積＞という形式で受信する方式等が  
考えられる。とにかく、画像情報よりも先にヘッダ情報  
として受信する必要がある。

【0031】この画像サイズ情報は、図2に示す基準符  
号量演算部21に送信され、そこにおいてブロック当り  
の基準（平均）の符号量を演算する。例えば、プリンタ  
の所有しているメモリ容量をA、入力した画像サイズを  
8×8のブロックに分割した場合のブロック数をBとす  
ると、 $A/B$ がブロック当りの基準（平均）の符号量に  
相当する。

【0032】画像情報は、I/O12から、1ブロック  
ライン分、バッファ13に格納され、圧縮処理部15の  
図3に示すのブロック化回路31に送信される。圧縮処  
理部15では、ブロック化回路31でブロック化した1  
ブロック分の画像情報をDCT回路32に送り、ここで  
DCT変換を施し、ジグザグスキャン回路33において  
ブロック内低周波域から高周波域に図4に示す様に、ジ  
グザグ状に一次元方向に並び換えられる。

【0033】続いて量子化部34で、符号量制御部から  
の量子化テーブルの情報に従って量子化処理が行われ、  
量子化係数はハフマン符号化35においてエントロピー  
符号化が施され、メモリ16に1ブロック分の符号が格  
納される。以上の説明において、符号化領域カウンタ1  
7は、ブロック化して符号化処理の施したブロックを加  
算していくアキュムレータより成っており、現在までの  
終了したブロック数が図2に示す掛け算器22に入力さ  
れる。

【0034】掛け算器22では、基準符号量演算部21  
からのブロック当りの基準となる符号量の情報と、符号  
化が終了したブロック数との積が演算される。例えば、  
前述した $A/B$ がブロック当りの基準符号量とした場合  
には、現在までKブロック終了していると仮定すると、  
 $(A \times K)/B$ の値が演算される。全てのブロックが終  
了した場合には $K=B$ となる為、掛け算器の出力はAと  
なる。

【0035】掛け算器の結果は、減算器23において、  
あらかじめ決めておいた下限設定値24が減算される。  
この掛け算器22、減算器23のそれぞれの出力は、現  
在までの符号化の終了した領域で基準となる符号量から  
ある値だけ下方向にオフセットを設定したものである。  
すなわち、下限設定値を $\beta$ とすると、現在まで、Kブ  
ロック終了していると仮定すると、減算器23からは  
 $((A \times K)/B) - \beta$ が出力される。

【0036】一方、メモリ16に格納されていく符号量  
は、符号量カウンタ18によつてカウントされていく。  
いま、Kブロック符号化が終了した場合のメモリに格納  
された符号量を $C_k$ とすると比較器25、26におい  
て、それぞれ $(A \times K)/B$ と $C_k$ の大小関係、 $((A$

10

20

30

40

50



$\times K) / B) - \beta$  と  $C_k$  との大小関係が比較される。判

$$(A \times K) / B \leq C_k \quad \dots 1$$

$$((A \times K) / B) - \beta \leq C_k < (A \times K) / B \quad \dots 2$$

$$C_k < ((A \times K) / B) - \beta \quad \dots 3$$

のいずれかにあるかを判定する。

【0037】この判定手段の結果により、もし前記の1の場合には、圧縮条件変換部28において、もう少し粗い量子化をする様に変更し、前記3の場合にはもう少し細かい量子化ができるように条件を変更する。前記2の場合には、基準となる符号量となる符号量にほぼ沿っている（許容幅に入っている）として、量子化条件は変更しない。

【0038】圧縮条件変更部28の情報は、圧縮処理部へと送信され、変更後の条件によつて、それ以降のブロックが符号化処理される。以上、説明した制御をわかりやすく示したのが図5である。図5において、横軸は累積符号化ブロック数を示し、最終了地点は、入力画像サイズ情報に基づき演算された全ブロック数を示している（この演算されたブロック数をBとおく）。

【0039】縦軸は累積符号量を示し、最終地点は、プリンタの所有メモリを示す（このプリンタの所有メモリをAとおく）。原点を通り、傾きがA/Bの直線が基準符号量の直線を示している。その直線よりも下限設定値βだけ下方に引かれた直線が前述した許容幅を示した直線になる。

【0040】すなわち、本実施例は、符号化していった軌跡をこの2本の直線間に押さえ込もうと制御するものである。このように制御することにより符号量の制御を1パスで実現できる。なお、当然ではあるが、2種の量子化テーブルは、符号量に差が表れるようにあらかじめ実験的に設定しておく必要がある。

【0041】以上説明した様に本実施例によれば、画像情報を格納するメモリを保有するプリンタにおいて、累積符号量と累積符号化領域の情報により、逐次、符号量を制御することにより、1回のパスで画像の可変長符号を一定のメモリ内に符号量を制御することができる。その結果、ホストコンピュータからの画像情報の転送時間が少なく済むし、可変長符号化を用いているため、画像情報の偏りに合った良好な圧縮処理が実現できる。

【0042】＜第2の実施例＞図6に本発明に係る第2の実施例の要部ブロック図を示す。第2の実施例における符号量制御を説明する図7も参照して本発明に係る第2の実施例を以下に説明する。第2の実施例は、第1の実施例の応用であり、上述した図2の符号量制御部のブロック図と同様構成には同一番号を付し詳細説明を省略する。

【0043】図中、61は基準符号量演算部を示し、第1の実施例同様に、ブロック当りの基準となる符号量を算出する。第2の本実施例では、ブロック当りの基準となる符号量の設定に特徴がある。第2の実施例では、上

\* 定手段27では、この比較結果を入力して、

$$\dots 1$$

$$\dots 2$$

$$\dots 3$$

限設定値62からの値（αとおく）を受け取り、所有メモリ容量Aよりもα分減算した値を基準となる最終了地点として設定する。すなわち、ブロック当りの基準符号量（図7の直線の傾きに相当する）を  $(A - \alpha) / B$  とする。

10 【0044】いま、符号化領域カウンタからの入力Kブロックだと仮定する。すると、掛け算器22の出力は  $((A - \alpha) \times K) / B$  となる。掛け算器の結果は、加算器63において、あらかじめ定めておいた上限設定値62との加算がなされ、また、減算器23においては、あらかじめ定めておいた下限設定値24（βとおく）が減算される。

【0045】この加算器63、減算器23のそれぞれの出力は、現在までの符号化の終了した領域で基準となる符号量からある値だけ、上方向と下方向にオフセットを設定したものである。すなわち、加算器63からは、

$$((A - \alpha) \times K) / B + \alpha, \text{ 減算器23からは、}$$

$$((A - \alpha) \times K) / B - \beta \text{ が出力される。}$$

【0046】いま、Kブロック符号化が終了した場合のメモリに格納された符号量を  $C_k$  とすると比較器27、28において、それぞれ、  $((A - \alpha) \times K) / B + \alpha$  と  $C_k$  の大小関係、  $((A - \alpha) \times K) / B - \beta$  と  $C_k$  との大小関係が比較される。判定手段27では、この比較結果を入力して、

$$((A - \alpha) \times K) / B + \alpha \leq C_k \quad \dots 1$$

$$30 \quad C_k < ((A - \alpha) \times K) / B + \alpha \text{ かつ}$$

$$((A - \alpha) \times K) / B - \beta \leq C_k \quad \dots 2$$

$$C_k < ((A - \alpha) \times K) / B - \beta \quad \dots 3$$

のいずれかにあるかを判定する。

【0047】第1の実施例同様に、この判定手段の結果により、もし、前記の1の場合には圧縮条件変更部28においてもう少し粗い量子化をする様に変更し、前記3の場合にはもう少し細かい量子化ができるように条件を変更する。前記2の場合には、基準となる符号量にほぼ沿っている（許容幅に入っている）として、量子化条件は変更しない。

【0048】圧縮条件変更部28の情報は、圧縮処理部へと送信され、変更後の条件によつてそれ以降のブロックが符号化処理される。すなわち、本実施例においては、基準となる符号量の上下に幅（オフセット）を有していること、また、基準符号量の算出が異なっているのが特徴であり、以上の様に制御することにより、よりブロックごとの符号量の誤りを許容できるような符号量制御が可能となる。

50 【0049】＜第3の実施例＞図8に本発明に係る第3の実施例の要部ブロック図を示す。第3の実施例におけ

る符号量制御を説明する図9も参照して本発明に係る第3の実施例を以下に説明する。第3の実施例では、基準符号量からのオフセット分の許容幅を設定するのではなく、直接、累積符号化ブロック数と累積符号量の関係から圧縮条件を設定する（第3の実施例では、量子化テーブルを5種類保持していると仮定する）。

【0050】図8において、81は量子化テーブル設定手段を示し、I/Oからの画像サイズ情報を入力することにより、全ブロック数（Bとおく）を算出する。また、符号化領域カウンタから、累積符号化ブロックの情報を受信することにより、累積の符号化ブロックの全ブロック数Bに対する比率も算出できる。すなわち、図9に示した様なグラフをROMの構成で、あらかじめ設定しておくことが可能である。

【0051】いま、累積符号化ブロックがKブロックであつたとする。また、保持している量子化テーブルは、量子化の粗い方からQ4、Q3、Q2、Q1、Q0とナンバーが付けられているとする。量子化テーブル設定手段81からの出力は<Q3の上限値>、<Q2の上限値>、<Q1の上限値>、<Q0の上限値>の4種の信号であり、それぞれ比較器82、83、84、85に送信される。

【0052】符号量カウンタからの入力がCkであつたとすると、それぞれの量子化テーブルの上限値がそれぞれの比較器にて比較される。Qnの上限値をQnUPとおくと、比較結果は、

$Q3UP < Ck$  ... 1 (Q4のテーブル)

$Q2UP \leq Ck < Q3UP$  ... 2 (Q3のテーブル)

$Q1UP \leq Ck < Q2UP$  ... 3 (Q2のテーブル)

$Q0UP \leq Ck < Q1UP$  ... 4 (Q1のテーブル)

$Ck < Q0UP$  ... 5 (Q0のテーブル)

の5通りに分類できる。

【0053】判定手段86は、比較結果を受けてどの量子化器の範囲に属しているかを判定する。量子化テーブル選択手段87において、前記判定手段86の判定結果に基づき、保持している量子化テーブルの中から適合するテーブルを選択し、圧縮処理部へと送信される。尚、以上の説明では、比較器を4個保有する構成を述べたが、累積符号量の変化は連続であることを考えれば、より、簡便な構成（例えば、比較器の数を減らした構成など）でも実現できる。

【0054】また、本実施例では、図9に示す様に、累積符号化ブロック数が増す程、各テーブルの切り換え範囲が狭くなる様な構成も可能である。すなわち、累積符号化ブロックが増える程、各ブロックごとの符号量の偏りが相殺される様になるためである。とりわけ、プリン

タにおいては、入力画像は、画像読み取り装置等から入力した自然画像、ホストコンピュータにより人工的に作成した画像、文字・線画像等の様々な画像情報が、接続したホストコンピュータ等から合成された形式で送信されることが多い。

【0055】そうした場合には、符号化のはじめの段階では許容幅を大きく設定して、符号量の偏りを取るようにして、符号化の広範にて符号量の制御していくようにすることが可能だ。以上、量子化条件の切り換えにより、符号量制御の例を述べたが、量子化器のテーブルは、復号時も容易に読みだせることが前提であり、量子化条件の切り代わったポイントで切り換え情報を符号化しても良いし、量子化テーブルをインデックス情報にして符号化しても良い。

【0056】また、前述したように、各々の量子化テーブルは、ある代表的な画像を用いて、ある符号量になるように、あらかじめ実験的に求めておくのが良い。また、本実施例では、ブロック単位で、符号量の検知を行っているが、数ブロック単位、ブロックライン単位での検知も当然、有効である。

<第4の実施例>以下、図面を参照して本発明に係る第4の実施例について詳細に説明する。

【0057】（1）第4の実施例における構成の説明  
図10において、801は第4の実施例における画像符号化装置を示すブロック図である。図において、802は入力画像信号をブロック化するブロック化部、803はブロックごとに系列変換を行なう変換部803、804は変換係数ブロックを量子化マトリクスを用いて量子化する量子化部、805は量子化された変換係数を可変長符号を用いて符号化する符号化部、806は符号化部805で符号化された符号量のある単位ごとに累積する符号量累積部、807は符号量累積部806に格納されている符号量から符号量の増加率を判定する増加率判定部、808は増加率判定部807での判定結果から閾値を切り換える閾値切り換え部、809は量子化部804を用いる量子化マトリクスを設定する量子化マトリクス設定部である。

【0058】（2）処理手順の説明

次に、以上の構成を備える第4の実施例における画像符号化装置801での画像符号化制御を図11に示すフローチャートに従って説明する。画像信号が入力されると図11の処理に進み、先ずステップS1で量子化マトリクス設定部809においてデフォルトの量子化マトリクスが設定される。次にステップS2で入力画像信号はブロック化部802で $n \times n$ 画素にブロック化され、ブロックの取り出しが行われる。

【0059】次にステップS3で変換部803において $n \times n$ 画素のデータに対して系列変換を行い、 $n \times m$ 個の変換係数を算出する。ここで、変換部803系列変換として離散コサイン変換を用いた場合における離散コサ



イン変換式は下記に示す数 1 の変換式を用いるのが望ましい。

$$F(\mu, \nu) = \frac{4}{n^2} C(\mu) C(\nu) \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} f(i, j) \cos(2i+1)\mu\pi/2n \times \cos(2j+1)\nu\pi/2n$$

$$\begin{aligned} \text{ここで} \\ C(\mu) &= \frac{1}{\sqrt{2}} & (\mu=0) & & C(\nu) &= \frac{1}{\sqrt{2}} & (\nu=0) \\ C(\mu) &= 1 & (\mu \neq 0) & & C(\nu) &= 1 & (\nu \neq 0) \end{aligned}$$

$f(i, j)$  : 入力画像データ  
 $F(\mu, \nu)$  : DCT 係数

【0061】次にステップ S 3 で、量子化部 804 において変換係数の位置ごとに変換係数を量子化マトリクスのそれぞれの値で割り、量子化を行う。ここで、量子化マトリクスのそれぞれの値が大きいと、量子化により「0」による係数が多くなり、符号量が少なくなる。続いてステップ S 5 で、符号化部 805 において量子化した係数と、量子化に用いたマトリクス情報を符号化する。そしてステップ S 6 で符号量累積部 806 において符号量を累積し、格納しておく。次にステップ S 7 で累積単位の処理が終了したか否かを調べ、累積単位分の処理が終了していなければステップ S 2 に戻り、以上のステップ S 2 ～ステップ S 6 の処理を 1 ブロック・ライン分繰り返す。

【0062】ステップ S 7 で累積単位の処理が終了した場合にはステップ S 8 に進み、増加率判定部 807 において前ブロック・ライン分の符号量と該ブロック・ライン分の符号量の差 D を、所定の閾値 TH1 と比較する。ここで、差 D が閾値 TH1 よりも大きければ、ステップ S 9 に進み、閾値切り換え部 808 において量子化マトリクス設定部 809 で用いる閾値テーブルを切り換えステップ S 10 に進む。一方、ステップ S 8 で差 D が閾値 TH1 よりも小さければ直ちにステップ S 10 に進む。

【0063】ステップ S 10 では、該ブロック・ラインまでの全累積符号量 C と現在設定されている閾値テーブルの該符号化領域での値 TH2 を比較する。そして累積符号量が閾値 TH2 よりも大きければ、ステップ S 11 に進み、量子化マトリクス設定部 809 においてこれまでよりもそれぞれの値が大きい量子化マトリクス（圧縮率大）を設定しステップ S 13 に進む。

【0064】一方、累積符号量が閾値 TH2 より小さければ、ステップ S 12 に進み、量子化マトリクス設定部 809 においてこれまでよりもそれぞれの値が小さい量子化マトリクス（圧縮率小）を設定しステップ S 13 に進む。ステップ S 13 では、該処理ブロックが最終ブロックであるか否かを判定する。処理ブロックが最終ブロックならば処理を終了し、処理ブロックが最終ブロックでなければステップ S 2 に戻り、以上の処理を最終ブロックまで繰り返す。

\* 【0060】  
 \* 【数 1】

【0065】なお、図 11 に示すステップ S 7 における判定は上述した 1 ブロック・ライン毎に行う例に限定されるものではなく、任意のブロック毎に行うことも当然に本実施例の範囲に含まれるものである。例えば、数ブロック・ラインを対象に行なってもよい。この場合には、ステップ S 8 では前ブロック・ラインと該ブロック・ラインの符号量の差 D と閾値 TH1 を比較するのではなく、最終的な目標の符号量に占める数ブロック・ライン分の符号量の割合 D' と、ある閾値 TH3 を比較する。

【0066】更に、図 11 に示すステップ S 8 の処理で用いる閾値 TH1 も、上述の例に限定されるものではなく、例えば画像の位置ごとに複数用意しておいてもよい。閾値 TH1 を画像の位置（符号化処理の段階）に応じて変化させることによりステップ S 9 での閾値の切り換えをより適応的に行なうことができる。更にまた、図 11 に示すステップ S 8 やステップ S 10 で用いる閾値 TH1, TH2 を、上述した様にあらかじめ画像の位置ごとのテーブルとして用意しておくのではなく、次式により画像の位置情報から計算で求めてもよい。

【0067】

TH1 = m × 画像の位置

TH2 = n × 画像の位置

このとき、m, n を切り換えることにより閾値の切り換えを行なうことができる。以上説明したように本実施例によれば、入力画像信号を複数画素からなるブロックに分割するブロック化部と、ブロックごとに系列変換する変換部と、変換された変換係数を量子化マトリクスを用いて量子化する量子化部と、量子化された変換係数を可変長符号を用いて符号化する符号化部と、符号化された符号量を累積する符号累積部と、符号量累積部の累積符号量を元に量子化部で用いる量子化マトリクスを切り換える量子化マトリクス設定部とを備え、更に、符号量累積部において累積された符号量が増加率を判定する増加率判定部と、増加率判定部での判定結果から量子化マトリクス設定部で量子化マトリクスの切り換えに用いる閾値を切り換える閾値切り換え部とを備えることにより、量子化マトリクスの切り換えに用いる閾値を画像の位置

における符号化情報量に応じて適応的に切り換え、画像全体での量子化幅を段階的に変化させることができる。

【0068】また、符号後の符号量を目標の値に固定することができる。

＜第5の実施例＞以下、図面を参照して本発明に係る第5の実施例を詳細に説明する。図12は本発明に係る第5の実施例における画像符号化装置を示すブロック図である。同図において、上述した図21に示す従来例と同様構成には同一番号を付し、詳細説明を省略する。即ち、401～409（除く404）は画像データ入力端子、バッファ、DCT変換部、パッキング処理部、圧縮データ出力端子、横サイズ入力端子、縦サイズ入力端子、符号量演算部を示している。第5の実施例における構成と上述した図21に示す構成における細部の異なる機能については後述する。

【0069】また本実施例では、さらに量子化後のデータを図13に示すスキャン順序に並び替えるスキャン変換部101、スキャン変換されたデータを、各データのエン트로ピーに基づいて符号化を行なうエン트로ピー符号化部102、該エン트로ピー符号化部102にて発生した全符号化量を計数するための符号量計数部103、1ブロックを符号化するたびに1つのパルスを入力するための入力端子104、符号量演算部409から出力される1ブロックの割り当て符号量を1ブロック符号化すると共に累積加算していくアキュムレータ105、符号量計数部103の出力からアキュムレータ105の出力を減算する減算器106、該減算器106の出力結果に基づき複数の量子化テーブルの切り換えを制御する量子化テーブル切り替え制御部107、複数の量子化テーブルを格納している量子化テーブル格納部108を備えている。

【0070】第5の実施例における画像符号化装置を用いた画像出力装置の構成としては、従来例の説明で使った図22の構成を用いることができる。従って、以下の説明は画像出力装置全体の説明は省略し、以上に説明した図12の構成における制御動作を説明する。コンピュータ等の外部装置（不図示）から送られてきたデータは、図22と同様、インタフェース部501にて画像のヘッダー情報と実際の画像データとに分けられ、ヘッダー情報中の画像の横サイズ情報は図12における端子407に入力される。一方、縦サイズ情報中の画像の横サイズ情報は、図12における端子407に入力される。縦サイズ情報は端子408に入力される。画像データは端子401に入力される。端子407、408に入力された画像の横サイズ、縦サイズ情報は符号量演算部409に送られ、1ブロックに割り当て可能なビット数が演算され、出力される。

【0071】一方、端子401に入力された画像データは、バッファ402に格納された後、ブロック単位で読み出されDCT変換部403に送られる。1ブロック分

の画像データをバッファ402からDCT変換部403へ送るたびに端子104にパルスを入力し、アキュムレータ105に送る。アキュムレータ105は、画像の符号化処理に先立ち、初期値「0」にクリアされ、端子104よりパルスを受けとるごとに、符号量演算部409の出力値を累積加算して結果を出力する。このアキュムレータの出力値120は、画像を符号化することによって発生した総符号量の基準となるものであり、以下基準符号量と呼ぶ。DCT変換部403に送られた画像データは、ブロック単位でDCT変換が行なわれ、次の量子化部404'で量子化される。

【0072】ここで、上述した図21に示す従来例では、DCT変換部の各周波数成分に対して、あらかじめ符号化ビット数が決められており、そのビット数に収まるよう量子化を行なう必要があつたが、本実施例では、情報量の少ないブロックは少ないビット数で符号化するため、各周波数成分に対してあらかじめ符号化ビット数を決めるようなことはしない。

【0073】そのかわり、各周波数成分に対して量子化ステップを定め、それに基づいて量子化を行なう。その量子化ステップは、量子化テーブル格納部108から量子化部404'に与えられる。量子化部404'で量子化されたデータは、スキャン変換部101に送られ、ここで図13に示す順序にスキャン変換される。一般に、自然画像では高域の周波数成分ほど電力が小さく、量子化後にゼロになりやすい。よって図13のような順序に並び替えられると、ゼロが後半に集中しやすくなる。このため、符号化・復号化の際にブロックの区切りが識別出来るようにすれば、後側に連続するゼロについては、符号化を省略することが可能となる。こうすることにより、1ブロックの符号化データが、ブロックの区切り情報のみ、あるいは、直流成分とブロックの区切り情報のみで済んでしまうといった場合が出てくる。

【0074】スキャン変換部101でのスキャン変換後の各周波数成分データは、エン트로ピー符号化部102において、各々の値の出現確率に応じて可変長符号に変換するエン트로ピー符号化を行なう。このエン트로ピー符号化によって発生した符号データ及び符号量は、それぞれパッキング処理部405、符号量計数部103へ送られる。

【0075】パッキング処理部405は、数ビット単位で発生する符号データをバイト単位もしくはワード単位の情報にまとめ、圧縮データとして端子406へ出力する。一方、符号量計数部103は、画像の符号化に先立ちゼロにクリアされ、エン트로ピー符号化部102から受け取った符号量を順次加算してゆき結果を出力する。

【0076】この符号量計数部の出力値121は、画像を符号化することによって実際に発生した符号量の各時点での総和であり、以下発生符号量と呼ぶ。該発生符号量121は、符号化する画像の性質によつて大きくもな

10

20

30

40

50



り、小さくもなるが、基準符号量120に近い値をとることが望ましい。発生符号量121が基準符号量120よりあまりにも大きいと、全画像データの符号化が完了する以前に記憶部503の空き領域が無くなってしまう。全画像データを保存することが出来なくなってしまう。

【0077】また、発生符号量121が基準符号量120よりあまりにも小さいと、記憶部503を有効に使っていないことになり、本来、低い圧縮率で画像の劣化を低くおさえることが可能であつたにもかかわらず、そうしていなことになる。そこで、第5の実施例においては、減算器106にて、発生符号量121から基準符号量120を減算して、差分値123を求め、該差分値123の絶対値が小さくなるように制御を行なう。

【0078】即ち、量子化テーブル切り換え制御部107では、該差分値123を複数の閾値と比較し、それらの閾値との大小関係によつて、複数の量子化テーブル内の1つを選択する選択信号124を量子化テーブル格納部108へ出力する。該量子化テーブル格納部108は、それぞれ圧縮率の異なる複数の量子化テーブルを有しており、該選択信号124により、その複数の量子化テーブル内の1つを選択して、量子化部404'で量子化される周波数成分に対応した量子化ステップ信号125を該量子化部404'へ出力する。

【0079】差分値123の絶対値が小さくなるようにするには、差分値123が大きくなつたときに、圧縮率の高い（発生する符号量の少ない）量子化テーブルへ切り換え、差分値123が小さく（マイナス方向に大きい）なつたときに圧縮率の低い量子化テーブルへ切り換えるようにすればよい。以上説明した様に第5の実施例では、発生符号量121から基準符号量120を減算器106にて減算し、その差分値123を求めておいてから、量子化テーブルの選択を行なう構成になつていますが、減算器106及び量子化テーブル切換制御部107の両方の機能を有するものをテーブルとして用意し、このテーブルに発生符号量121、基準符号量120を入力すると量子化テーブルの選択信号124が出力として取り出せるようにしてもよい。

【0080】このように、符号化処理によつて発生した符号量（発生符号量）と基準となる符号量（基準符号量）とに基づいて圧縮率を制御すると、記憶部を有効に使用することができるため、該記憶部の容量を従来より少なくすることができ、コストの安い画像符号化装置、ひいては、該画像符号化装置を内蔵したコストの安い画像出力装置を実現できる。

【0081】また、記憶部を有効に使用することができるよということを別の観点から見ると、限られた記憶容量で再現可能な最高の画質に近い画像が保存できるということであり、該装置の高性能化にもつながる。以上説明した様に第5の実施例によれば、符号化処理により発

生した符号量を計数して得られる発生符号量と、基準符号量生成手段から得られる基準符号量に基づいて、圧縮率の制御を行なうことにより、画像の情報に偏りがある場合にも、記憶部を有効に使用することが出来る。このため、記憶部の容量を数十%減らすことができ、記憶部のコストを大幅に下げることが出来る。

【0082】特に以上の符号化装置は圧縮した画像を格納する記憶部を有する画像出力装置に好適であり、該画像出力装置の低コスト化に寄与できる。

10 <第6の実施例>図14は、本発明に係る第6の実施例における画像符号化装置の構成を示すブロック図である。図14において、上述した第5の実施例における図12と同様構成には同一番号を付している。第6の実施例では、第5の実施例と比し、画像出力装置が1度に出

20 力できる1ページの画像が、それぞれ性質の異なる複数の小画像から成る場合に対応するものである。  
【0083】符号化する順序は各々の小画像を1つずつ順番に行なうものとする。本実施例では、第5の実施例における量子化テーブル切換制御部107の替わりに、量子化ステップ幅の制御を行なうためのスケールファクタ切換制御部201を備え、該スケールファクタ切換制御部201から出力されるスケール値を、量子化テーブル格納部108から出力される値に掛け合せるための乗算器202を使用している。

【0084】又、第1の実施例において、画像の横サイズ情報と縦サイズ情報を入力していた端子407、408には、各々複数画像の属性情報（画像サイズ情報を含む）と複数画像の識別情報を入力することにする。1つの画像の中でも領域により情報の偏りがあることはすでに述べたが、性質の異なる画像、例えばコンピュータグラフィックス等で生成した画像とスキヤナ等で読み込んだ画像の間でも、高域周波数成分に大きな差がある。よつて、それらの画像を同じ符号量の割り当てで符号化したのでは、高域周波数成分を多く持つ高画質な画像が大きく劣化してしまうことになる。そこで、このような場合、各画像の入力源や自己相関等のパラメータを用いて、符号量の割り当てに重みを付ければ、圧縮による各画像の劣化をバランスさせることができる。

40 【0085】そこで、第6の実施例では、画像の符号化に先立ち、複数の画像の各々の属性情報を入力端子407より入力し、符号量演算部409にて、各画像ごとに所定の画像データ量に対する割り当て符号量を計算する。例えば、A、Bという2つの画像があり、各々のデータ量がDa、Dbブロックあり、割り当て符号量の重みをWa、Wbとし、記憶部の容量をMビットとすると、画像Aに対して割り当てられる1ブロック当たりの符号量は、 $W_a \cdot M / (W_a \cdot D_a + W_b \cdot D_b)$  ビットとなり、画像Bに対して割り当てられる1ブロック当たりの符号量は、 $W_b \cdot M / (W_a \cdot D_a + W_b \cdot D_b)$  ビットとなる。



【0086】そこで、前記2つの式の共通項である $F = M / (W_a \cdot D_a + W_b \cdot D_b)$ をあらかじめ計算しておいて、画像Aを符号化する際には $W_a \cdot F$ を、画像Bを符号化する際には、 $W_b \cdot F$ を、符号量演算部409より出力する。画像A、Bのどちらを符号化中であるかを知るには、上述したインタフェース部501にて、画像データの転送を管理し、どちらの画像が転送中であるのかを認識し、その情報を端子408に入力すればよい。

【0087】符号量演算部409は、その情報をもとに、各画像に対応した割り当て符号量を出力する。実際には、画像データの転送と符号化タイミングの間には多少の時間的なずれがあるので、そのずれはインタフェース部501または符号量演算部501で吸収する必要がある。まず最初に、画像Aが転送され、それを符号化するものとする。符号量計数部103及びアキュムレータ105は、リセット信号（不図示）により初期値ゼロにクリアされる。画像Aの転送、並びに符号化が始まると、符号量演算部409は、 $W_a \cdot F$ という割り当て符号量をアキュムレータ105に出力する。1ブロックを符号化するごとに、端子104からパルスが入力され、アキュムレータ105は、 $W_a \cdot F$ という値を累積加算していく。

【0088】一方、画像データは、第1の実施例と同様、バツファ402にてブロック化されDCT変換部403、量子化部404'、スキャン変換部101を経て、エントロピー符号化部102にて符号化される。エントロピー符号化部102にて符号化された符号量が符号量計数部103で計数される。ここで、第6の実施例においては、上述した第5の実施例と異なり、量子化部404'で用いられる量子化ステップは、量子化テーブル格納部108の出力にスケールファクタ切替制御部201から出力されるスケール値を乗算器202にて乗算して作られる。

【0089】従つて、符号化開始直後においては、減算器106の2つの入力ゼロであるためその出力も勿論ゼロとなる。このため、結局スケールファクタ切替制御部201にはゼロが入力され、それに対応したスケール値が乗算器202に入力される。そして、符号化が進むにつれて、アキュムレータ105出力の基準符号量120と、符号量計数部103の出力の発生符号量121との間に差が生じ、減算器106の出力にその差分値123が現われる。

【0090】第5の実施例と同様、本実施例においても、差分値123の絶対値が小さくなるようにスケールファクタの制御が行なわれる。即ち、差分値123が大きくなったらスケール値を大きくして量子化ステップを粗くすることにより発生符号量を少なくし、逆に差分値123が小さくなったらスケール値を小さくして量子化ステップを細くすることにより、発生情報量を増や

す。

【0091】以上説明した動作内容で、画像Aの符号化が行なわれ、該画像Aの符号化が終了すると、次は画像Bが転送され符号化される。画像Bの符号化に先立ち、符号量計数部103及びアキュムレータ105をリセット信号（不図示）により、ゼロクリアすると共に、符号量演算部409にて $W_b \cdot F$ という割り当て符号量を演算し、アキュムレータ105に出力する。以後、画像Bの符号化は、画像Aの符号化と同様に行なわれる。

【0092】このように、第6の実施例によれば、複数の画像を符号化する場合においても、発生符号量と基準符号量とに基づいて圧縮率を制御することにより、記憶部を有効に使用することができるため、該記憶部の容量を従来より少なくすることが出来る。なお、第6の実施例では、スケールファクタを制御することにより量子化条件を変更する構成について示したが、本発明は以上のレインに限定されるものではなく、第5の実施例で述べたような、量子化テーブルを切り換える方法であつてもかまわないし、他の方法であつてもよい。

【0093】以上の説明では、画像Aの符号化が終了して、画像Bの符号化を始める際に、符号量計数部103及びアキュムレータ105をゼロにクリアしたが、このクリアは無くてもよい。このクリアが無いと、画像Aの符号化が終了した時点で、発生符号量が基準符号量より小さかった場合、その差の記憶容量を画像Bの符号化の際に有効に活用できる。しかし、逆に発生符号量が基準符号量より大きかった場合、その分だけ画像Bを符号化する際に使用できる記憶容量が減ってしまうことになる。

【0094】＜第7の実施例＞本発明に係る第7の実施例における画像符号化装置のブロック構成を図15に示す。図15において、図12及び図14と同様構成には同一番号を付して詳細説明を省略し、主に相違する点について説明する。第7の実施例においては、所定の画像データ量に対する発生符号量がいつも基準符号量以内に収まるよう圧縮率の制御を行なうものである（以下では、所定の画像データ量をKブロックとして話を進める）。

【0095】即ち、Kブロックの画像データのある量子化条件の元で符号化した際に発生した符号量が、基準符号量をオーバーしてしまった場合、量子化条件を発生符号量が減る方向に変更して、再度、同じKブロックの画像データを符号化するわけである。2度目の符号化で発生した発生符号量が基準符号量以下であれば、該Kブロックの画像データの符号化は終了して、次のKブロックの画像データの符号化を行なう。しかしながら2度目の符号化で発生した発生符号量が基準符号量以下でなければ、さらに量子化条件を変更して、3度目の符号化を行なう。以下、発生符号量が、基準符号量以下になるまで、量子化条件を変更しながら繰り返し、符号化を行な

うわけである。

【0096】第7の実施例において、以上説明した処理を行なう場合、これまで説明した第5、第6の実施例のように、わざわざ符号量演算部409とアキュムレータ105を用いて基準符号量を生成する必要が無く、記憶部の容量、符号化する画像の全データ量、並びに該KブロックのKの値から、基準符号量を演算することができる。

【0097】そこで図15に示す第7の実施例では、基準符号量演算部301にて基準符号量を演算して出力する（信号線120は基準符号量となる）。該基準符号量の演算に必要な情報は端子302より入力する。また、第7の実施例では、基準符号量と発生符号量との差分値は必要ではなく、それらの値の大小関係が解ればよい。このため、比較器303にてその関係を判定する。その他の構成は、基本的に第6の実施例と同じである。但し、制御方法が多少異なるブロックがいくつか存在するが、以下の動作説明でその都度説明することにする。

【0098】以下、図15の構成を備える第7の実施例の動作を説明する。まず、最初にこれから符号化する画像の全データ量等、Kブロックに対する基準符号量を演算するのに必要な情報が、端子302を通して基準符号量演算部301に入力される。そして、該基準符号量演算部301にて、基準符号量が演算され、その結果が信号線120に出力される。一方、画像データは、第5、第6の実施例と同様にバッファ402にてブロック化され、DCT変換部403、量子化部404'、スキャン変換部101を経てエントロピー符号化部102にて符号化される。

【0099】そして、エントロピー符号化部102にて符号化された符号量が符号量計数部103で計数される。該符号量計数部103は、符号化処理に先立ち、勿論初期値としてゼロにクリアされている。またこの時、スケールファクタ制御部201も初期状態にリセットされており、初期状態に対応したスケール値を出力している。そして、乗算器202は、量子化テーブル格納部108の出力にそのスケール値を乗算し、結果を量子化部404'に送る。

【0100】Kブロックの画像データの符号化が済んだところで、符号量計数部103から出力される発生符号量121と基準符号量120を比較器303にて比較し、どちらの値が大きいか判定する。もし、発生符号量121が基準符号量120以下であれば、該Kブロックの画像データの符号化を終了し、次のKブロックの画像データの符号化を開始する。

【0101】逆に発生符号量121が基準符号量120より大きい時は、その情報をスケールファクタ切換制御部201に送り、再度該Kブロックの画像データをバッファ402より読み出し、符号化を行なう。そこでバッファ402は、一度読み出された画像データが再度読み

出されても大丈夫なように、しばらく該画像データを保持するといった制御が必要になる。スケールファクタ切換制御部201では、受けとつた情報から、以前出力していたスケール値よりも大きなスケール値を出力し、それにより、発生符号量を以前より少なくする。

【0102】こうして、再度Kブロックの画像データを符号化し、発生した符号化データはパッキング処理部405を経由して、記憶部503に格納する。第7の実施例においては、この格納の際に、第5、第6の実施例とは異なる制御が必要となる。即ち、同一の画像データを2度、3度と符号化して、発生した符号化データをシーケンシャルに記憶部503へ格納していったのでは、記憶部503に不要なデータが格納され、記憶容量が一杯になってしまう。

【0103】そこで、同一の画像データを複数回符号化した際には、2度目以降の符号化データは、1度目の符号化データの格納場所へ上書きすることで、不要なデータが残らないように制御する必要がある。このため、第7の実施例においては、符号化され格納されたデータ量（発生符号量）が、基準符号量より大きい場合、再度スケール値を大きくして、再符号化を行ない、発生符号量が、基準符号量以下になるまで繰り返す。

【0104】なお、第7の実施例において、KブロックのKの値は任意であるが、 $K=1$ の場合というのは従来例にて発生した問題と同様のことが、また問題になる。よって、該問題を改善するには、 $K>1$ である必要がある。

＜第8の実施例＞図16に本発明に係る第8の実施例における画像符号化装置のブロック構成を示す。第8の本実施例においては、符号量計数手段として、記憶部503を構成する1ユニットであるアドレス発生器を用いた場合の構成例である。

【0105】従って、これまで述べた第5～第7の実施例では、符号量計数手段として、記憶部をまったく含まない、符号化処理部のみのブロック図として説明してきたが、本実施例においては、該記憶部503を含むブロック図で説明する。該記憶部503は、図16において、破線で囲まれた領域に対応し、符号化データを格納するメモリ601、メモリ601にアドレス信号を送るアドレス発生器602、アドレス発生器602の制御、及びメモリ601の制御信号を生成するメモリ制御部603より構成される。

【0106】端子604は、データを伸張する際に、メモリ601から読み出した符号化データを出力する端子である。すでに述べたように、符号量計数手段としてアドレス発生器を用いるため、第8の実施例では、上述した図12、図14及び図17に示すような符号量のみを専用に計数する符号量計数部103が不要となる。その他の構成ユニットは、図12に示す第5の実施例と同様であるため、同一番号を付し、詳細説明を省略する。ま



た、符号化装置全体の動作内容も、第1の実施例とまったく同じであるため、説明を省略し、以下では、アドレス発生器602に関してさらに詳しく説明する。

【0107】第8の実施例の如く、符号量計数手段として、アドレス発生器602が使用可能な理由は、該アドレス発生器602から出力されるアドレス情報が、メモリの使用量、即ち、符号化データの発生量を表すことができるからである。実際に、アドレス情報が符号化データの発生量を表すには、該アドレス値は、ゼロからスタートして、1つつアドレスが増加するという動作が前提になり、第8の実施例も、この動作を前提としている。

【0108】ところで、本実施例のアドレス値の発生順序は上述の順序以外にもいろいろある。例えば、アドレス値を大きい方から順に発生させることもできる。このような場合には、該アドレス値を反転した値が符号化データの発生量を表すので、該反転値を符号量計数手段の出力とすればよい。また、メモリの0番地から、L-1番地までが、他の情報を格納する領域として確保されていて、L番地以降から順に符号化データを格納する場合には該アドレス値からLを減算した値を符号量計数手段の出力とすればよい。

【0109】以上説明したように、上述の各実施例によれば、符号化処理により発生した符号量を計数して得られる発生符号量と、基準符号量生成手段から得られる基準符号量に基づいて、圧縮率の制御を行なうことにより、画像の情報の偏りがある場合にも、記憶部を有効に使用することが出来る。よって、記憶部の容量を数十%減らすことができ、記憶部のコストを大幅に下げることが出来る。

【0110】また、圧縮した画像を格納する記憶部を有する画像出力装置に好適であり、該画像出力装置の低コスト化に寄与できる。

＜第9の実施例＞図17は本発明に係る第9の実施例における画像符号化装置の構成を示すブロック図である。同図において、上述した第5実施例～第8実施例の図12～図16と同様構成には同一番号を付し詳細説明を省略する。

【0111】第9の実施例では、特有のユニットとして、符号量計数部103にて計数した発生符号量を記憶部の容量で割り算する第1の除算器154、符号化する1画素もしくは1ブロックの画像データごとに1つのパルスが入力されるパルス入力端子155、該入力端子155よりのパルスをカウントして符号化した画像データ量を計数する画像データ計数部156、符号化する全画像のデータ量を入力する入力端子157、画像データ計数部156の出力を全画像データ量で割り算する第2の除算器158、第1の除算器154の出力から第2の除算器158の出力を減算する減算器159、該減算器159の出力結果に基づく、複数の量子化テーブルの切り

換えを制御する量子化テーブル切換制御部160を備えている。

【0112】第9の実施例においても、画像出力装置の全体構成は上述した第5実施例～第7実施例と同様で良い。以下、以上の構成を備える第9の実施例の動作説明を行なう。コンピュータ等の外部装置（不図示）から送られてきたデータは、上述実施例と同様、インターフェース部501において画像のヘッダ情報と実際の画像データとに分けられ、ヘッダ情報中の画像のサイズ情報は全画像データ量を示す数値に変換された後、入力端子157に入力されると共に、画像データは入力端子401に入力される。

【0113】入力端子401に入力された画像データは、上述実施例と同様バッファ402、DCT変換部403、量子化部404'、スキャン変換部101を経てエントロピー符号化部102にて符号化される。符号化されたビット単位のデータは、パッキング処理部にてバイト単位もしくはワード単位のデータにパッキングされ、端子406'に出力された後、記憶部503に格納される。

【0114】この時、バッファ402から画像データが読み出されるたびに入力端子155にパルスが入力され、画像データ計数部156に送られる。該画像データ計数部156は、画像データの符号化に先立ち、リセット信号（不図示）によりゼロにクリアされ、その後に入力されるパルスをカウントする。カウントされた値は第2の除算器158に送られ、入力端子157から入力された画像サイズ情報で除算される。

【0115】その結果、第2の除算器158からは全画像データ量に対する符号化済み画像データ量（符号化中の画像データも含む）の割合が出力される。符号量計数部103は、上述した様に画像の符号化に先立ち、ゼロにクリアされ、エントロピー符号化部102から受けとった符号量を順次加算してゆき、結果を出力する。第9の実施例では、符号量計数部103よりの出力値は第1の除算器154にて、符号化データを格納する記憶部503の容量で割り算され、該記憶部の使用割合が求められ、出力される。

【0116】ここで、第1の除算器154から出力される記憶部の使用割合は、前記第2の除算器158から出力される符号化済み画像データ量の割合（以下、前者を第1の割合、後者を第2の割合と呼ぶ）に近い値をとることが望ましい。すなわち、第1の割合が第2の割合よりあまりにも大きいと、全画像データの符号化が完了する以前に、記憶部503の空き領域が無くなってしまう、全画像データを保存することが出来なくなってしまう。

【0117】また、第1の割合が第2の割合よりあまりにも小さいと、記憶部503を有効に使っていないことになり、本来、低い圧縮率で画像の劣化を低く抑えるこ



とが可能であるにもかかわらず低く押さえていないことになる。この点を解決するため、第9の実施例では減算器159にて、前記第1の割合から第2の割合を減算して差分値170を求め、差分値170の絶対値が小さくなるように制御を行なう。

【0118】このため、量子化テーブル切換制御部160では、差分値170を受けとり、それを複数の閾値と比較し、それらの閾値との大小関係によつて、複数の量子化テーブルの内の1つを選択する選択信号124を量子化テーブル格納部108へ出力することになる。以後の動作は上述実施例と同様である。なお、上述実施例でも同様であるが、復号化時に使用する量子化テーブルは、符号化時に使用した量子化テーブルと同一のものでなければならない。よつて符号化時に量子化テーブルを切り換えた場合、復号化時にもまったく同じタイミングで量子化テーブルを切り換える必要がある。

【0119】そこで、前記量子化テーブル切換制御部160にて量子化テーブルの切換えがなされた場合、その切換え情報を信号線173を通してエントロピー符号化部120へ送り、該切換え情報を符号化し、圧縮符号化された画像データと共に記憶部に保存する。以上説明した制御方法で、全画像データを圧縮符号化すると、第2の除算器158から出力される第2の割合（符号化済み画像データの割合）は、最終的に“1”という値になる。それに対して、第1の除算器154から出力される第1の割合（記憶部の使用割合）は、最終的に“1”という値の近傍に落ち付くことになる。“1”の近傍ということは“1”を越える場合があることを意味し、該第1の割合が“1”を越えることは、記憶部を全部使いきってしまうと容量が足りなくなることを意味する。

【0120】よつて、前記第1の割合が最終的に“1”を越さないよう、符号化の後半の方では減算器159出力の差分値が0以下になるよう量子化テーブル切換制御部160の制御方法を変えていく。そのためには、符号化がどこまで済んだかを該量子化テーブル切換制御部160で知る必要があるので、信号線174にて第2の除算器158からの符号化済み画像データの割合を、量子化テーブル切換制御部160に送る。

【0121】以上説明した様に第9の実施例によれば、符号化データを格納する記憶部の容量に対する発生符号量割合と、全画像データ量に対する符号化済み画像データ量の割合に基づいて圧縮率を制御することにより、記憶部を有効に使用することができ、該記憶部の容量を数十%減らしても、十分に動作させることができ、記憶部のコストを大幅に下げることができる。

【0122】＜第10の実施例＞図18は本発明に係る第10の実施例における画像符号化装置の構成を示すブロック図である。第10の実施例は、全画像データ量に対する符号化済み画像データ量の割合を第9の実施例とは異なる構成で求めるものである。又、第9の実施例に

において、除算器154から出力される記憶部の使用割合が最終的に“1”を越さないように信号線174を付加し、量子化テーブル切換制御部160に、符号化済み画像データの割合を知らせていたが、該信号線174を使用しなくても記憶部が途中で足りなくなるようなことがないような構成とした。よつて、第10の実施例では前記信号線174が不要になった点、及び全画像データ量に対する符号化済み画像データ量を求める方法が異なる以外は、第9の実施例とまったく同じであり、同一ユニットには同一部番を付し説明を省略する。

【0123】以下、第10の実施例に固有の構成ユニットについて説明する。図18において、161は入力端子157から入力された画像データ量の逆数を演算する逆数器、105は入力端子104からパルスが入力される度に逆数器161の出力値を累積して加算する図12に示す第5実施例と同様のアキュムレータ、163はアキュムレータ105から出力される値に所定の値を乗算する乗算器である。

【0124】次に、以上の構成を備える第10実施例の動作説明を行なう。コンピュータ等の外部装置から送られてきたデータは、インターフェース部501にて画像のヘッダ情報と画像データとに分けられ、該画像データについては、第9の実施例と同様に、バツファ402、DCT変換部403、量子化部404'、スキャン変換部101を経てエントロピー符号化部102にて符号化される。符号化されたビット単位のデータは、パツキング処理部405にてバイト単位もしくはワード単位のデータにパツキングされ、端子406に出力された後、記憶部503に格納される。符号化部で発生した符号量は、すべて符号量計数部103にて計数された後、除算器154にて記憶部の容量で割り算され、記憶部をどれだけ使用したかを示す数値（記憶部使用割合）が出力される。

【0125】一方、画像のヘッダ情報に含まれる画像サイズ情報は符号化する画像の全データ量を示す数値に変換された後、入力端子157に輸入される。入力端子157に輸入された数値は、逆数器161にて、該数値の逆数に変換され、全画像データに対する1画素の割合が求められる。この数値はアキュムレータ162に輸入され、符号化する画像データの1画素ごとに入力されるパルス（入力端子156より入力される）によつて累積加算される。

【0126】アキュムレータ162の出力値は、全画像データに対する符号化済み画像データの割合であり、全画像データを符号化した後は勿論“1”という値になる。第9の実施例では、該符号化済みデータの割合なる数値を直接、減算器159に輸入していたので、除算器154の出力値（記憶部の使用割合を表わす）が、“1”を越さないような制御が必要であつた。

【0127】本実施例では、アキュムレータ162と減

算器159の間に乗算器163を配し、該乗算器163にて“1”より小さな値 $\alpha$ を乗ずることにより、除算器154の出力値が“1”を越えないようにする。第9の実施例の動作説明で述べた様に、乗算器163の最終出力が $\alpha$ であれば、除算器154からの最終出力は $\alpha$ の近傍になることが分かる。よって、該[ $\alpha$ の近傍]の最大値が1以下になるように $\alpha$ を設定すれば、符号化済み画像データの割合に応じて量子化テーブル切換の制御方法を変える必要がなくなる。

【0128】以上、説明した動作に基づいて、画像データの符号化を行えば、記憶部を有効に使用した圧縮が可能となり、全画像データを確実に符号化できる。なお、本実施例において、乗算器163は、アキュムレータ162と減算器159の間に配したが、アキュムレータ262と逆数器161の間に配してもよい。その際、逆数器161及び乗算器163における演算は、1つの画像に対して各々1回の演算で済むため、演算スピードは遅くてもよい。よって、それらの演算は専用回路で行なう必要がなく、他の制御に使用されている不図示のCPU等を用いて演算してもよい。また、除算器154は、記憶部の容量が2のべき乗の場合、単なるビットシフト回路に置き替えることが出来る。

【0129】尚、本発明は、複数の機器から構成されるシステムに適用しても、1つの機器から成る装置に適用しても良い。また、本発明はシステム或は装置にプログラムを供給することによって達成される場合にも適用できることは言うまでもない。

【0130】

【発明の効果】以上、説明したように、本発明によれば、画像情報を格納するメモリを保有するプリンタにおいて、累積符号量と累積符号化領域の情報により、逐次、符号量を制御することにより、1回のパスで画像の可変長符号を一定のメモリ内に符号量を制御することができる。

【0131】その結果、ホストコンピュータからの画像情報の転送時間が少なくて済むし、可変長符号化を用いているため、画像情報の偏りに合った良好な圧縮処理が実現できる。その結果、大容量を必要とするプリンタ、特に、カラープリンタにおいても少量の所有メモリで製品化を実現できる。

【0132】また、量子化マトリクスの切り換えに用いる閾値を画像の位置における符号化情報量に応じて適応的に切り換え、画像全体での量子化幅を段階的に変化させることができ、また、符号後の符号量を目標の値に固定することができる。更に、符号化処理により発生した符号量を計数して得られる発生符号量と、基準符号量生成手段から得られる基準符号量に基づいて、圧縮率の制御を行なうことにより、画像の情報に偏りがある場合にも、記憶部を有効に使用することが出来る。よって、記憶部の容量を数十%減らすことができ、記憶部のコスト

を大幅に下げることが出来る。また、圧縮した画像を格納する記憶部を有する画像出力装置に好適であり、該画像出力装置の低コスト化に寄与できる。

【図面の簡単な説明】

【図1】本発明に係る一実施例の画像出力装置の構成を示すブロック図である。

【図2】図1の符号量制御部の詳細構成を示す図である。

【図3】図1の圧縮処理部の詳細構成を示す図である。

10 【図4】本実施例の一般的なジグザグスキャンの説明図である。

【図5】本実施例の制御をグラフ化した図である。

【図6】本発明に係る第2の実施例の構成を示すブロック図である。

【図7】第2の実施例の制御をグラフ化した図である。

【図8】本発明に係る第3の実施例の構成を示すブロック図である。

【図9】第3の実施例の制御をグラフ化した図である。

20 【図10】本発明に係る第4の実施例における画像符号化処理装置を示すブロック図である。

【図11】第4の実施例での画像符号化制御を示すフローチャートである。

【図12】本発明に係る第5の実施例における画像符号化装置の構成を示すブロック図である。

【図13】第5の実施例におけるDCT変換後の各周波数成分をスキャン変換部で並び替える際の順序を示す図である。

【図14】本発明に係る第6の実施例における画像符号化装置の構成を示すブロック図である。

30 【図15】本発明に係る第7の実施例における画像符号化装置の構成を示すブロック図である。

【図16】本発明に係る第8の実施例における画像符号化装置の構成を示すブロック図である。

【図17】本発明に係る第9実施例における画像符号化装置の構成を示すブロック図である。

【図18】本発明に係る第10実施例における画像符号化装置の構成を示すブロック図である。

【図19】従来の技術を示すブロック図である。

40 【図20】従来例による画像符号化装置の構成を示すブロック図である。

【図21】本実施例に適用可能な画像符号化装置の好適な応用例である画像出力装置を示すブロック図である。

【図22】一般的なDCT変換後の各周波数成分を符号化する際のビット配分を示す図である。

【符号の説明】

11, 401, 407, 408 入力端子

12 プリンタのI/O

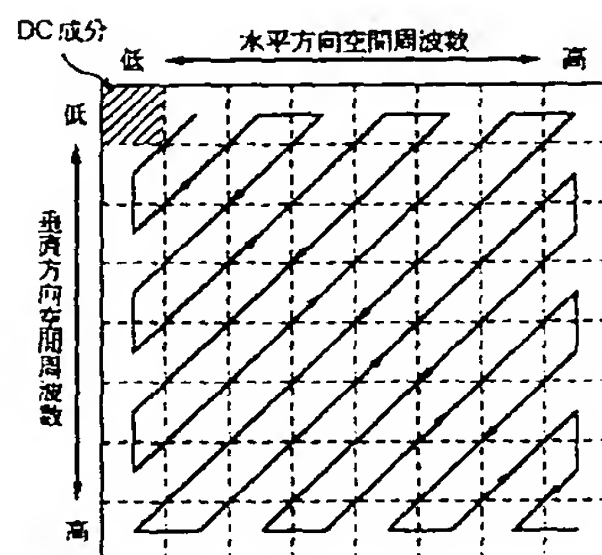
13 バッファメモリ

14 符号量制御部

50 15, 502 圧縮処理部

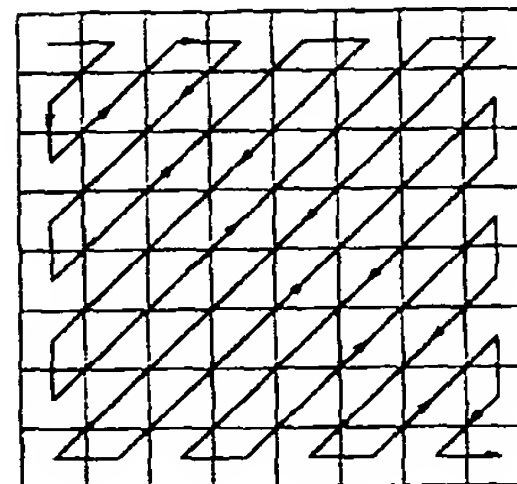
16, 601 メモリ  
 17 符号化領域カウンタ  
 18 符号量カウンタ  
 19 伸張処理部  
 20 プリンタエンジン  
 21, 61 基準符号量演算部  
 22 掛け算器  
 23, 106, 159 減算器  
 24 下限設定値  
 25, 26, 82~85, 303 比較器  
 27, 86 判定手段  
 28 圧縮条件変換部  
 31 ブロック化回路  
 32 DCT回路  
 33, 1006 ジグザグスキャン回路  
 34, 404, 2004 量子化部  
 35 ハフマン符号化  
 62 上限設定値  
 63 加算器  
 81 量子化テーブル設定手段  
 87 量子化テーブル選択手段  
 101 スキャン変換部  
 102 エントロピー符号化部  
 103 符号量計数部  
 105, 162 アキュムレータ  
 107 量子化テーブル  
 108 量子化テーブル格納部  
 154, 158 除算器  
 156 画像データ計数部  
 160 量子化テーブル切替制御部  
 161 逆数器  
 163, 202 乗算器

【図4】



201 スケールファクタ切替制御部  
 301 基準符号量演算部  
 402 バツファ  
 403 DCT変換部  
 404 量子化部  
 405 パツキング処理部  
 409 符号量演算部  
 410 ビット配分演算部  
 501 インターフェース部  
 10 502 圧縮処理部  
 503 記憶部  
 504 伸長部  
 505 画像出力部  
 602 アドレス発生器  
 603 メモリ制御部  
 801 画像符号化制御装置  
 802 ブロック化部  
 803 変換部  
 804 量子化部  
 20 805 符号化部  
 806 符号量累積部  
 807 増加率判定部  
 808 閾値切り換え部  
 809 量子化マトリクス設定部  
 1000 フレームメモリ  
 1001 DCT処理  
 1002 量子化器  
 1003 パスカウンタ  
 1004, 1008 スイッチ  
 30 1005 量子化テーブル  
 1009 符号量測定器  
 1010 スケーリングファクタ変更部

【図13】

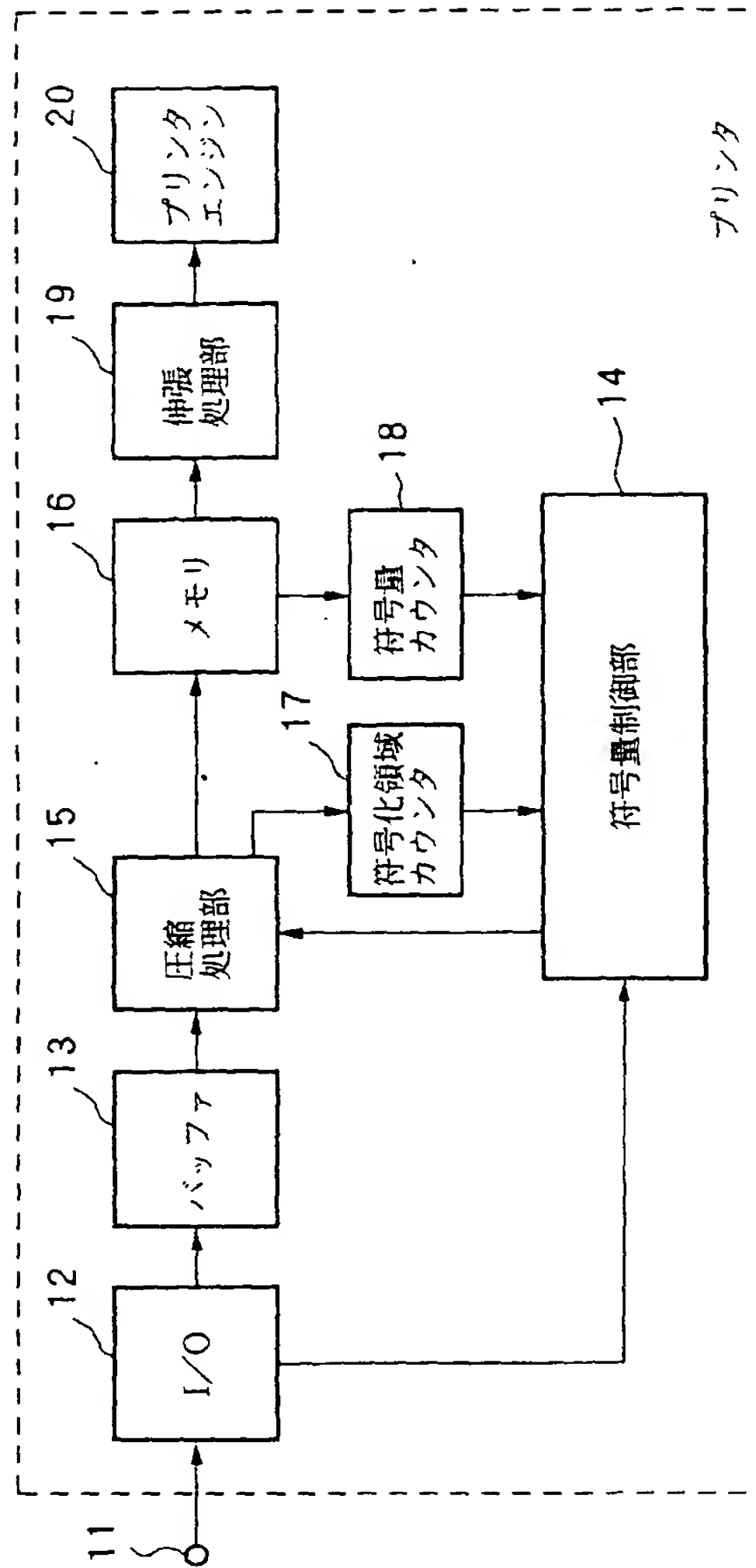


【図22】

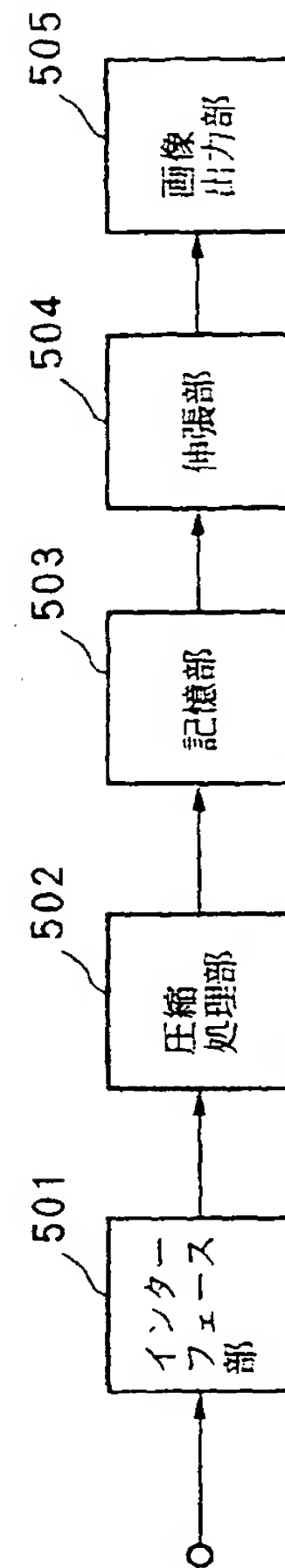
6	6	6	5	4	3	2	2
6	6	5	4	3	2	2	
6	5	4	3	2	2		
5	4	3	2	1			
4	3	2	1				
3	2	1					
2	1						
1							



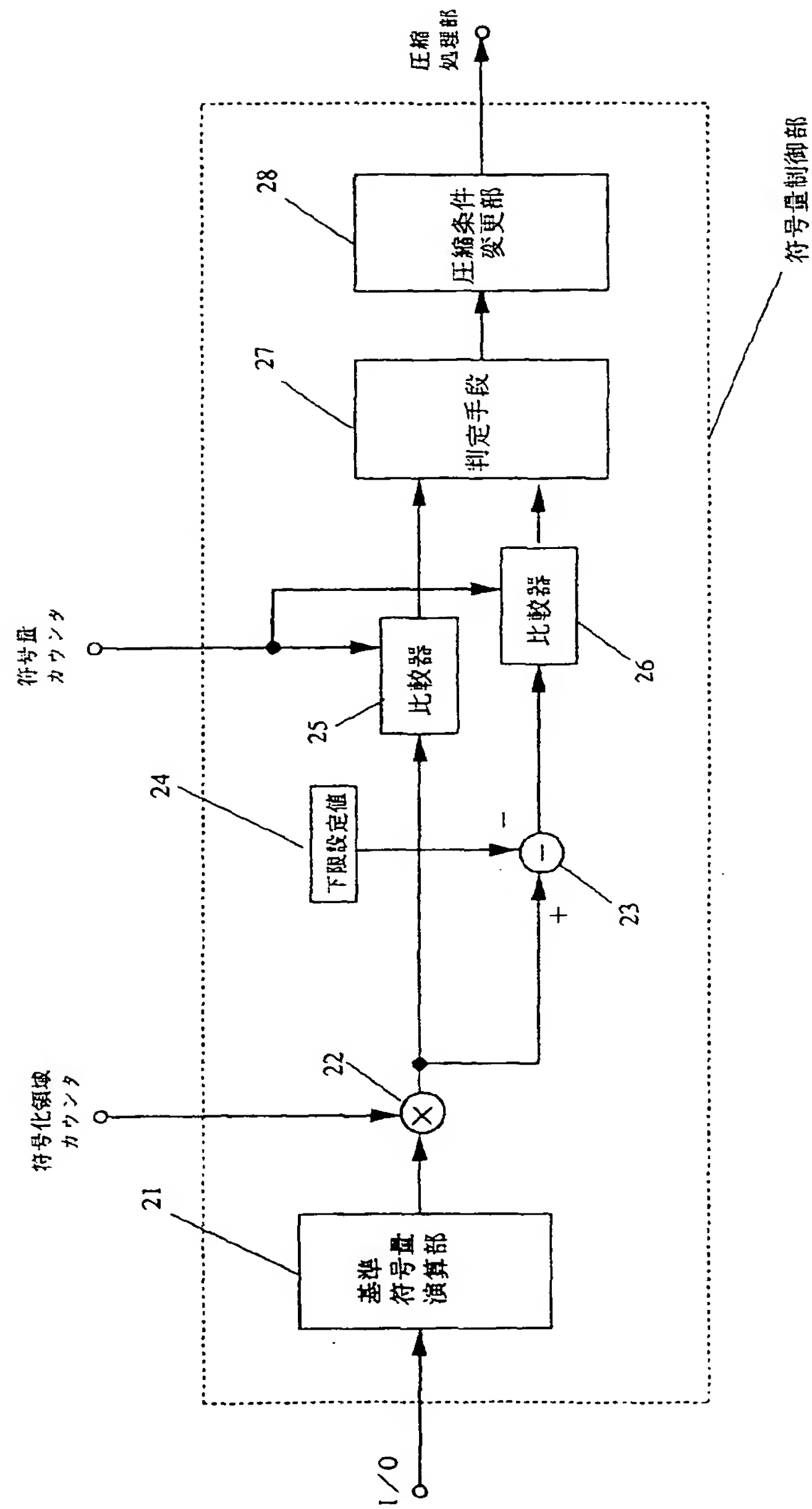
【図1】



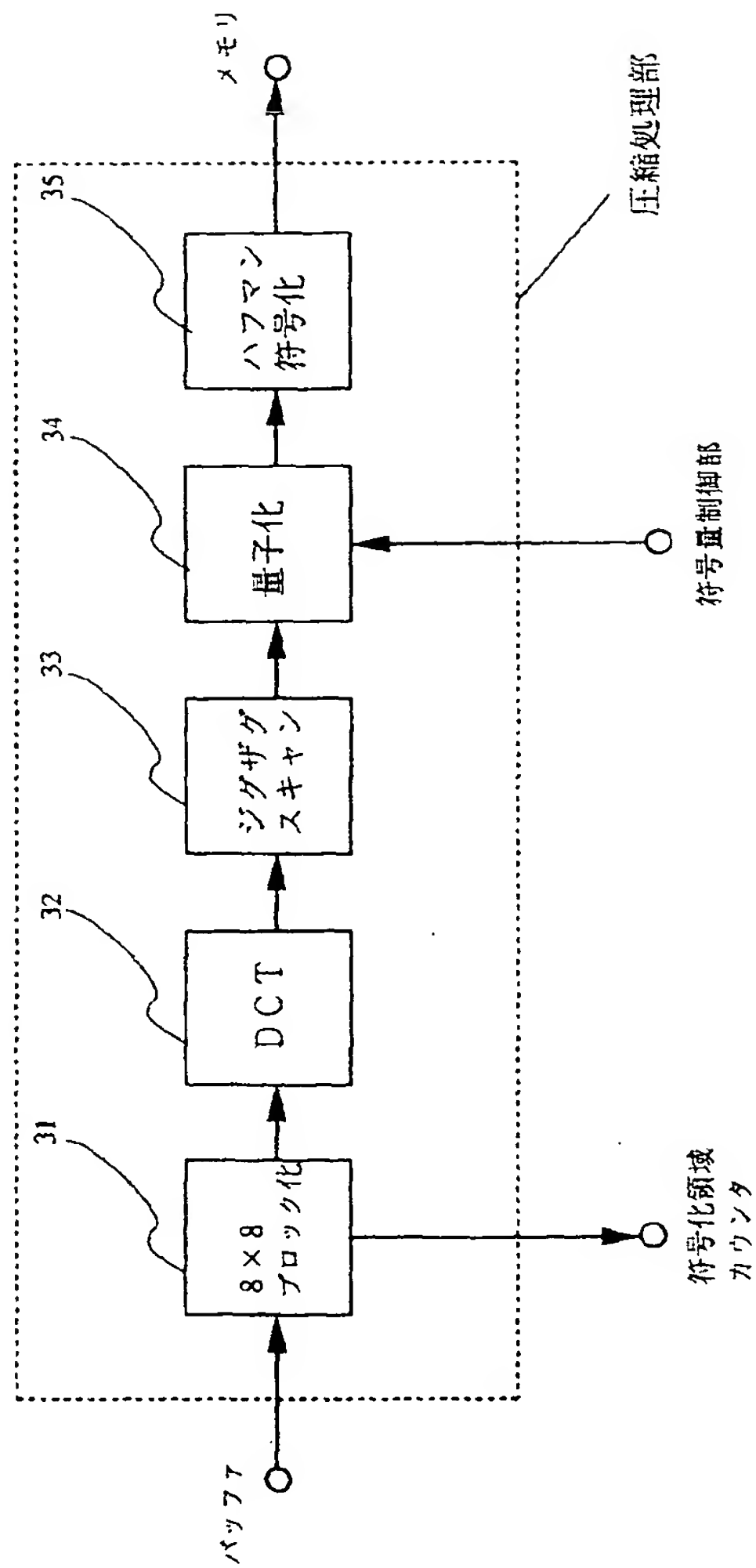
【図21】



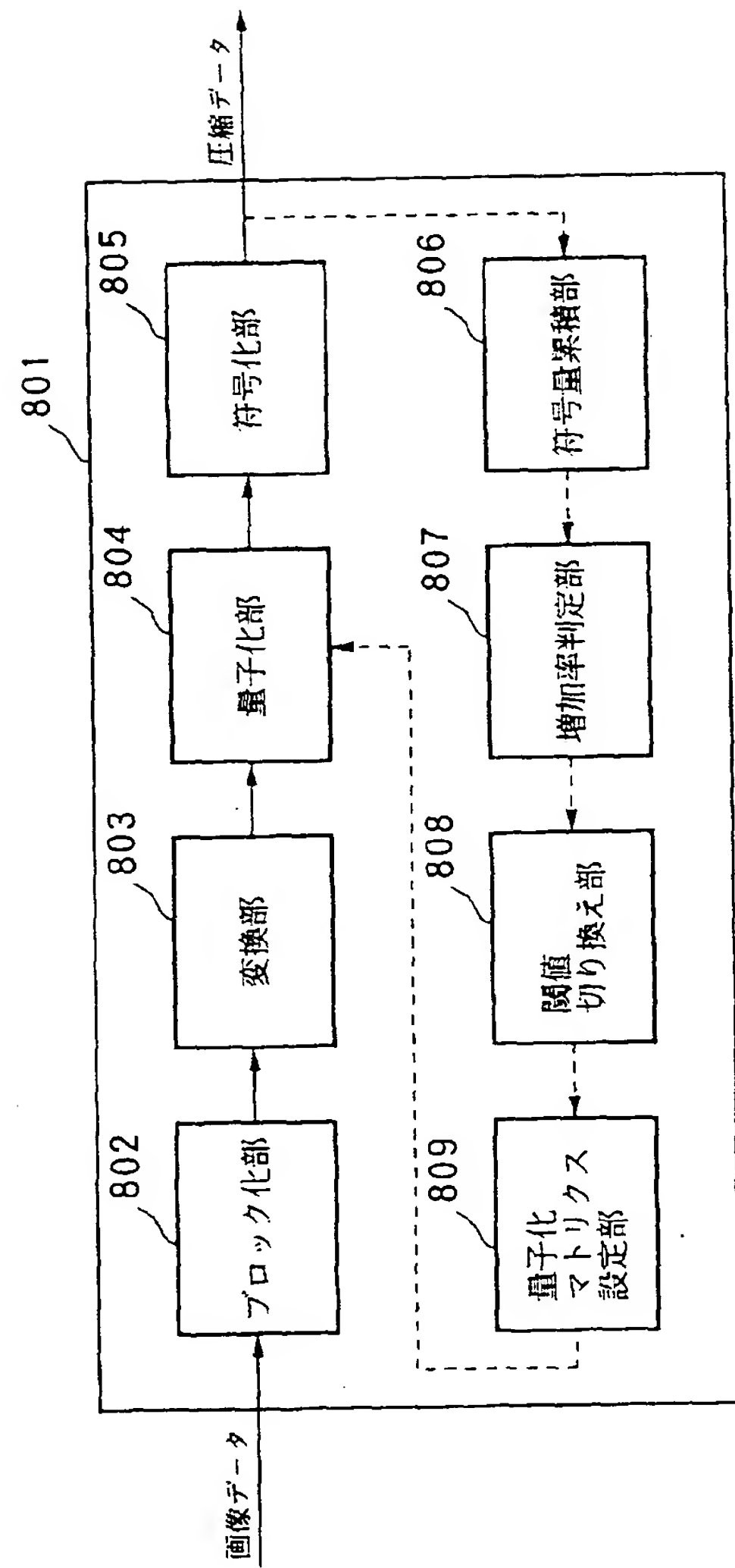
【図2】



【図3】

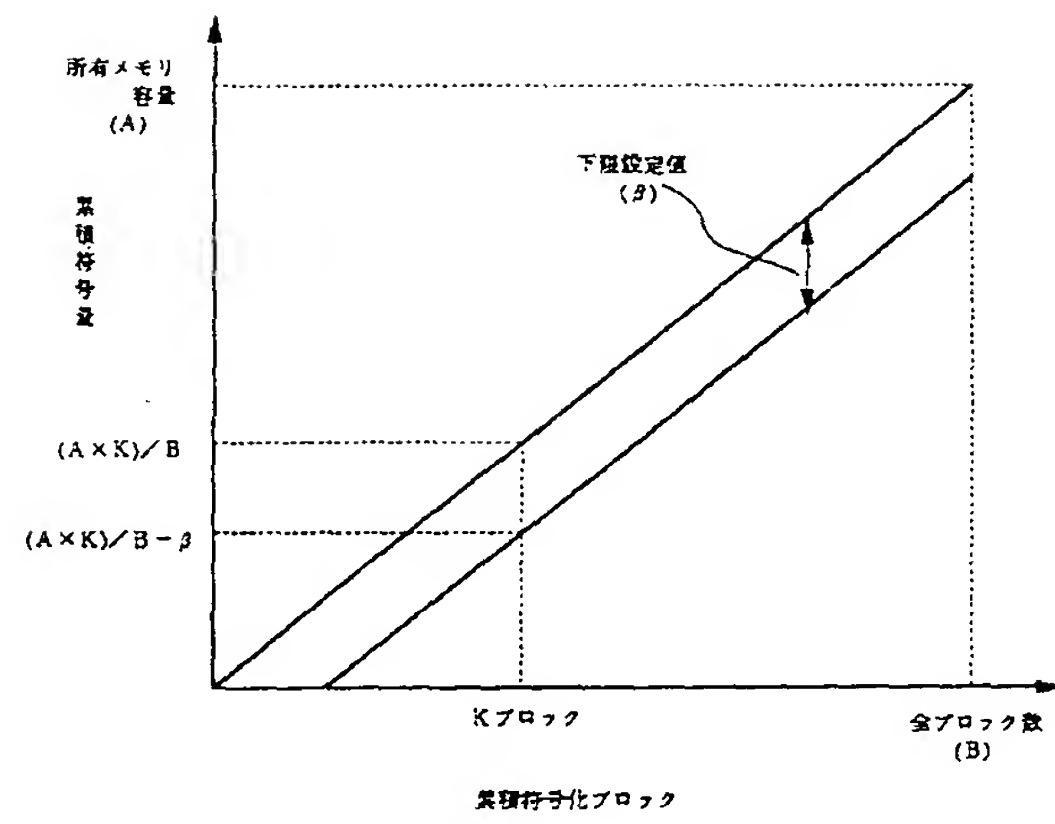


【図10】

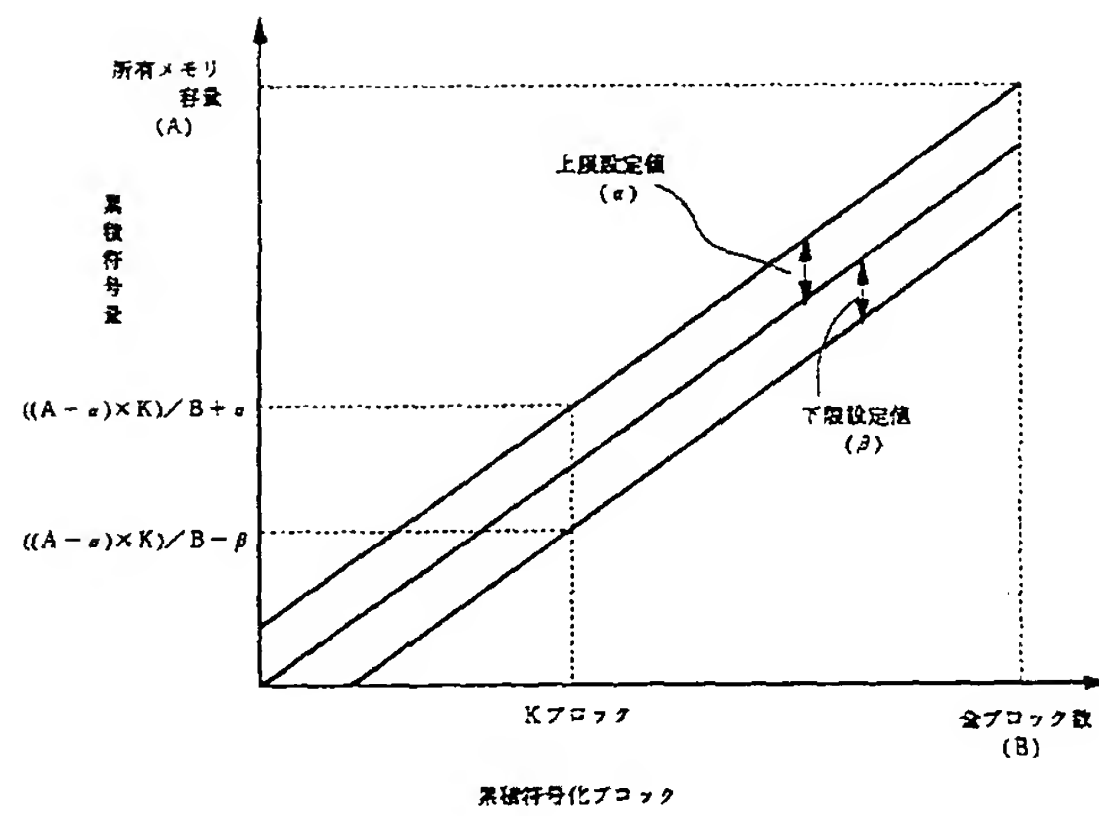




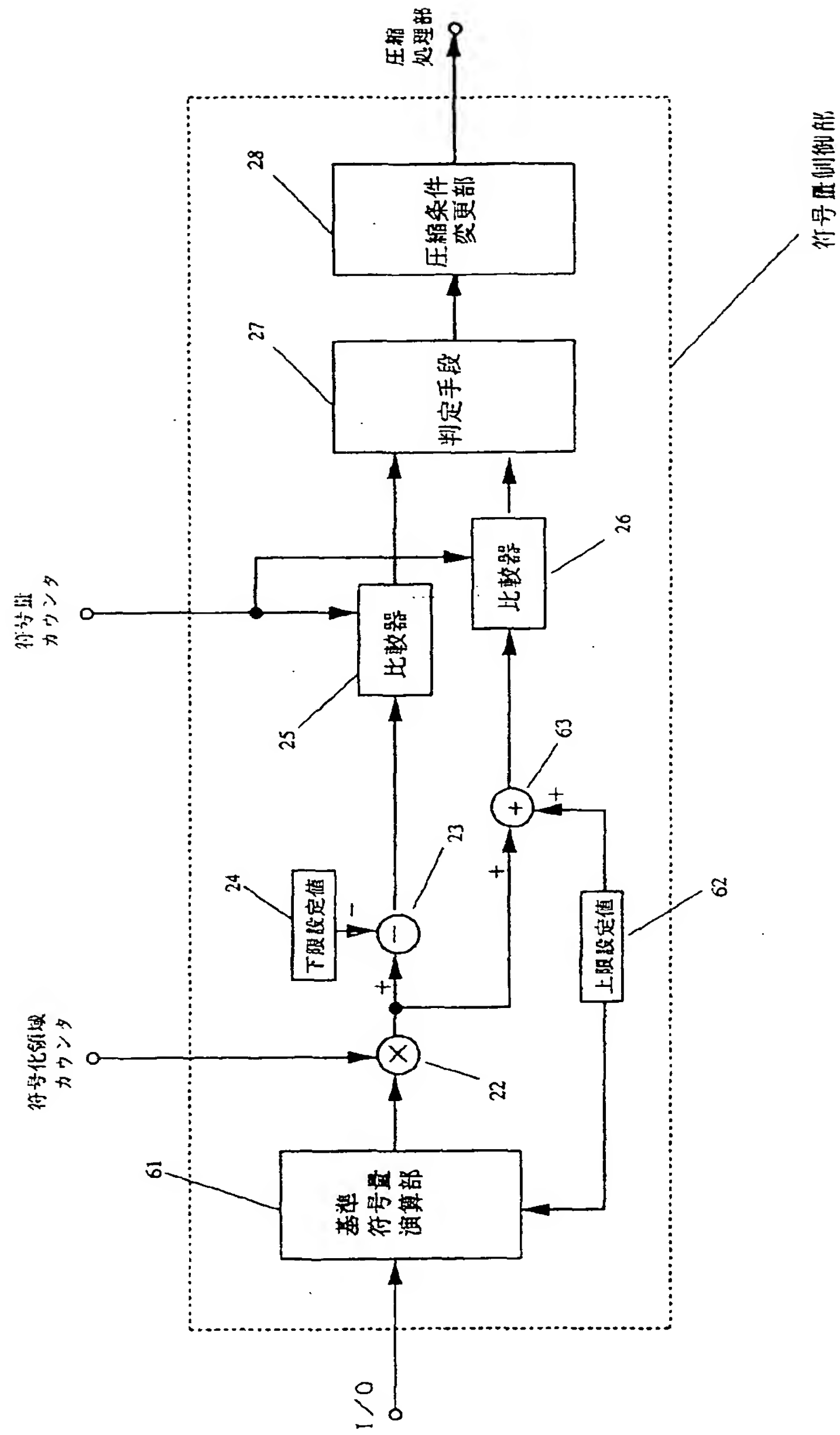
【図5】



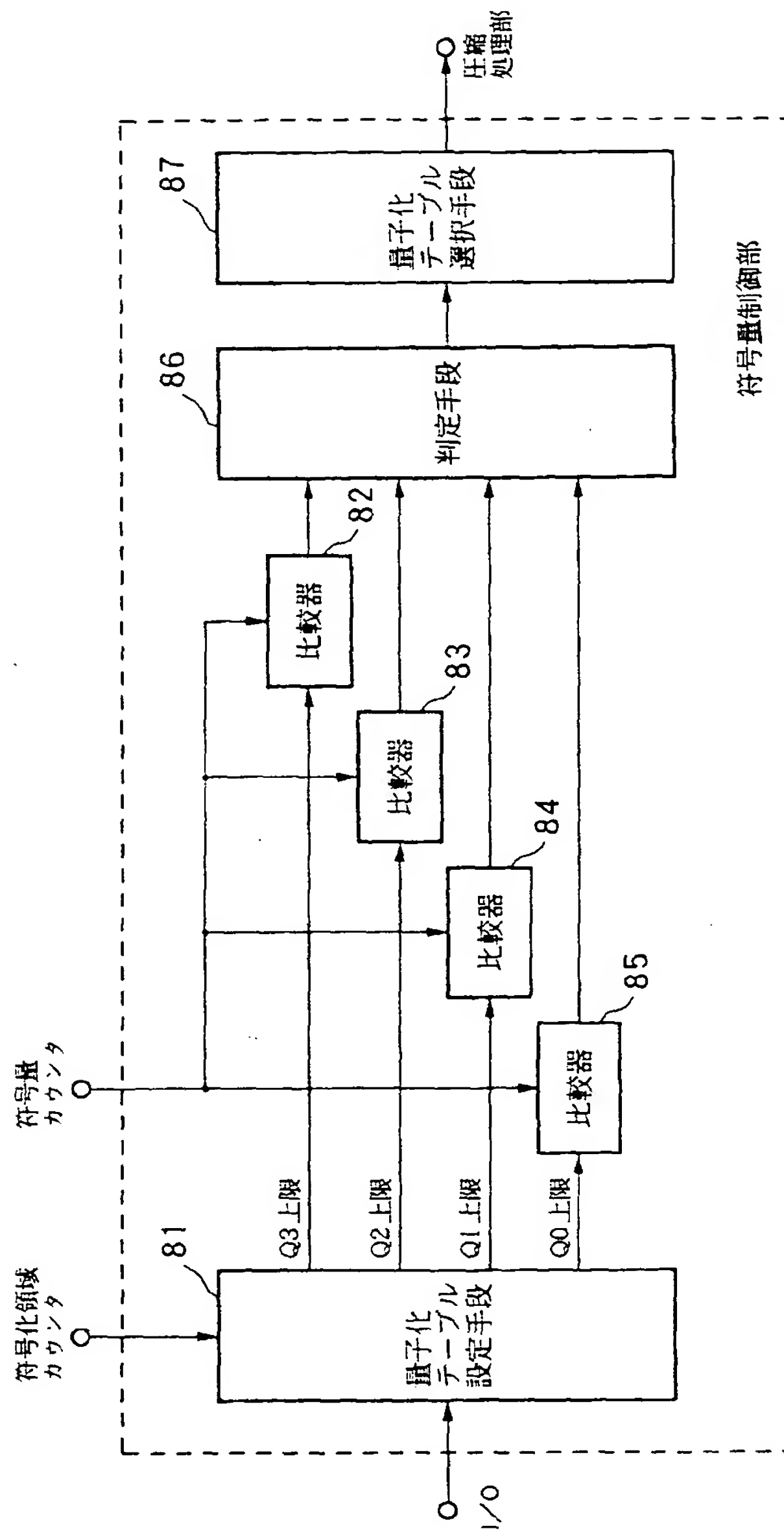
【図7】



【図6】

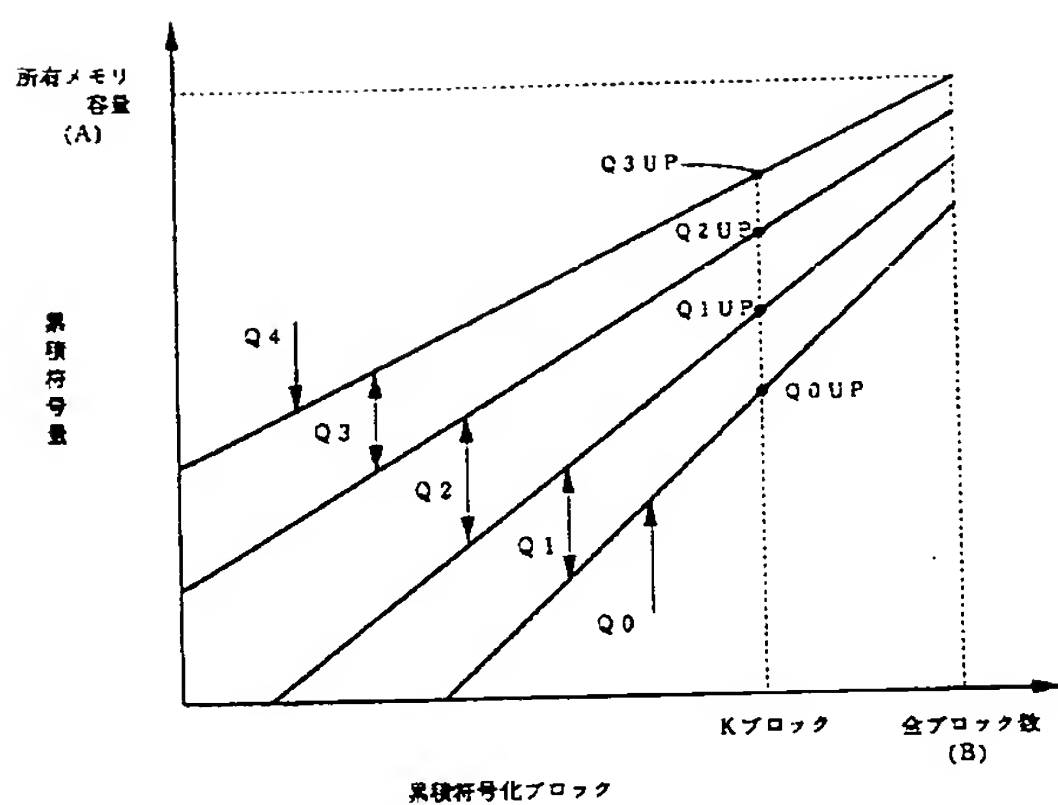


【図 8】

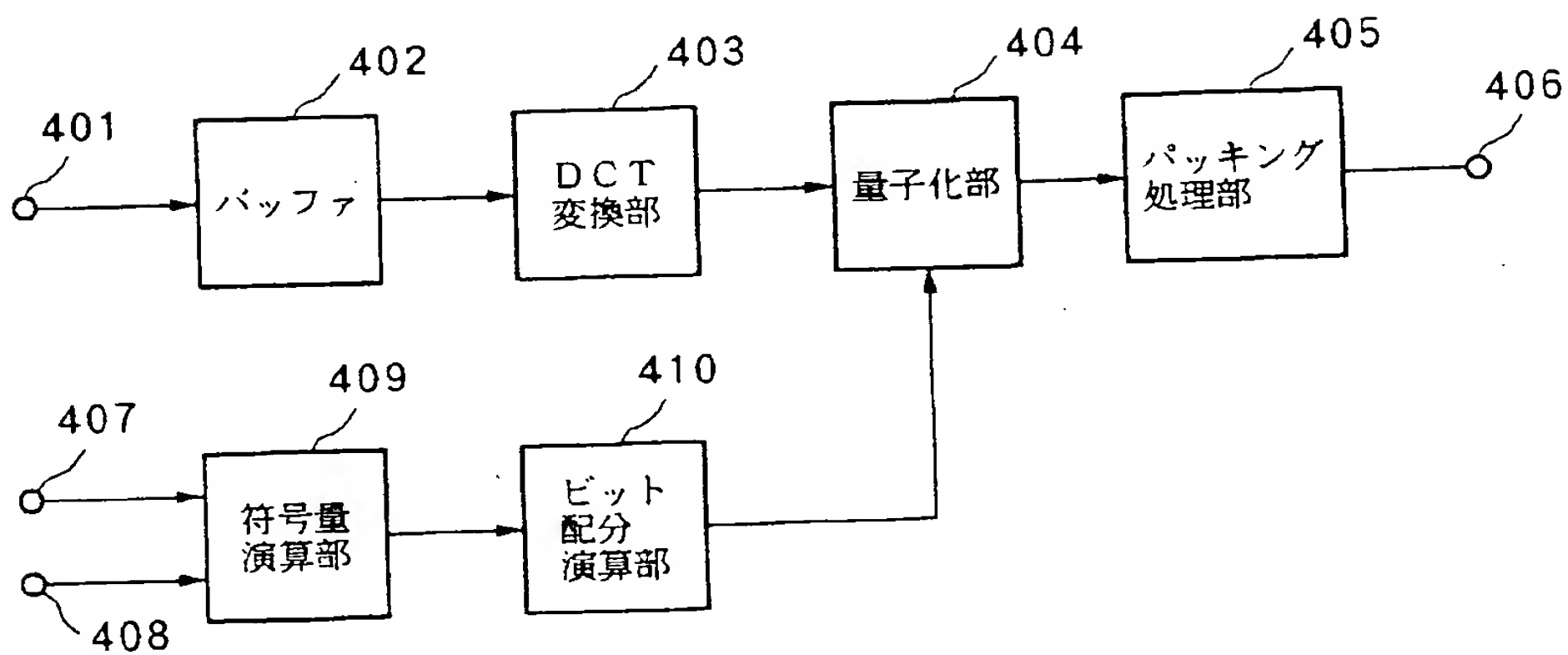




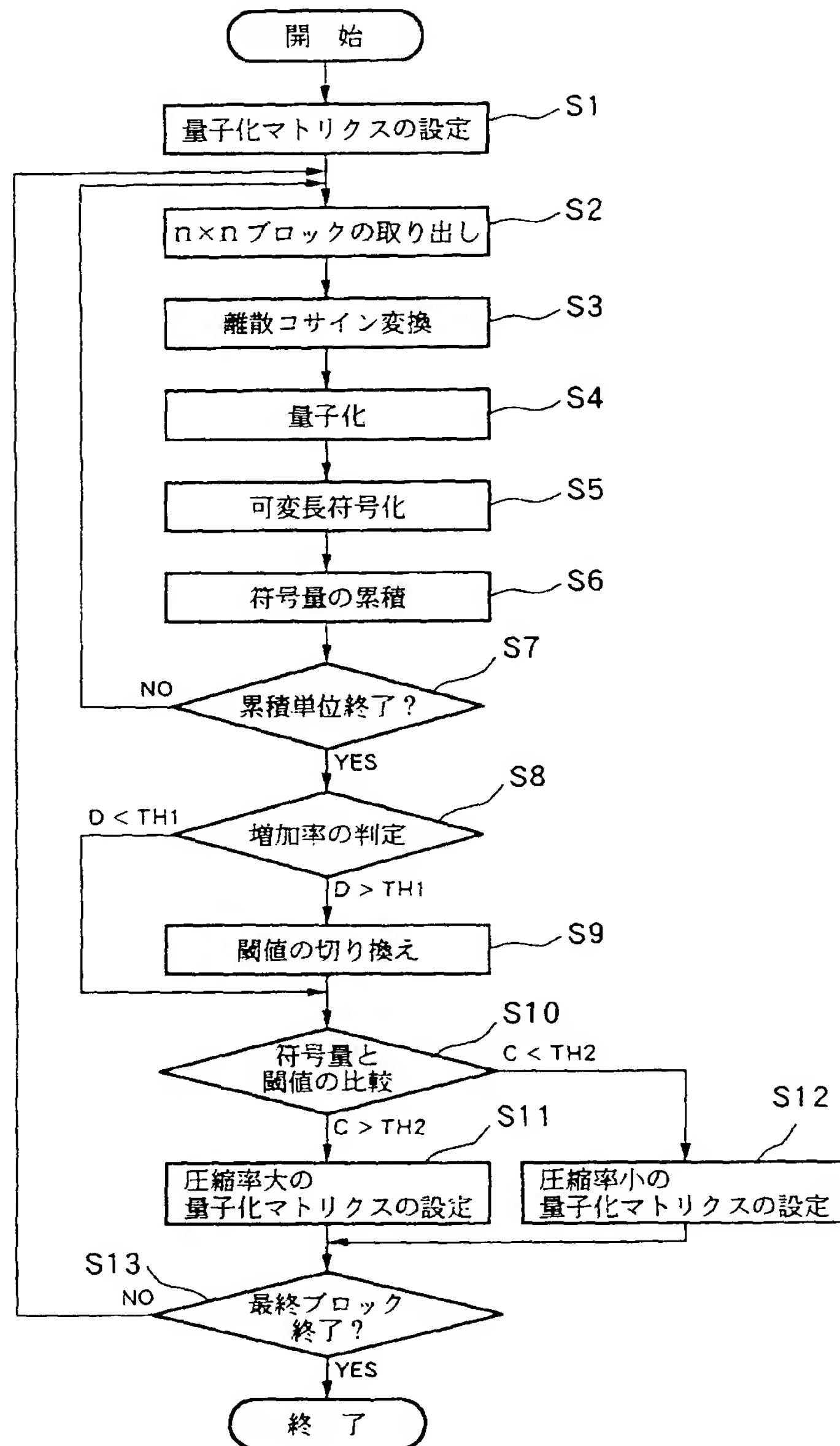
【図9】



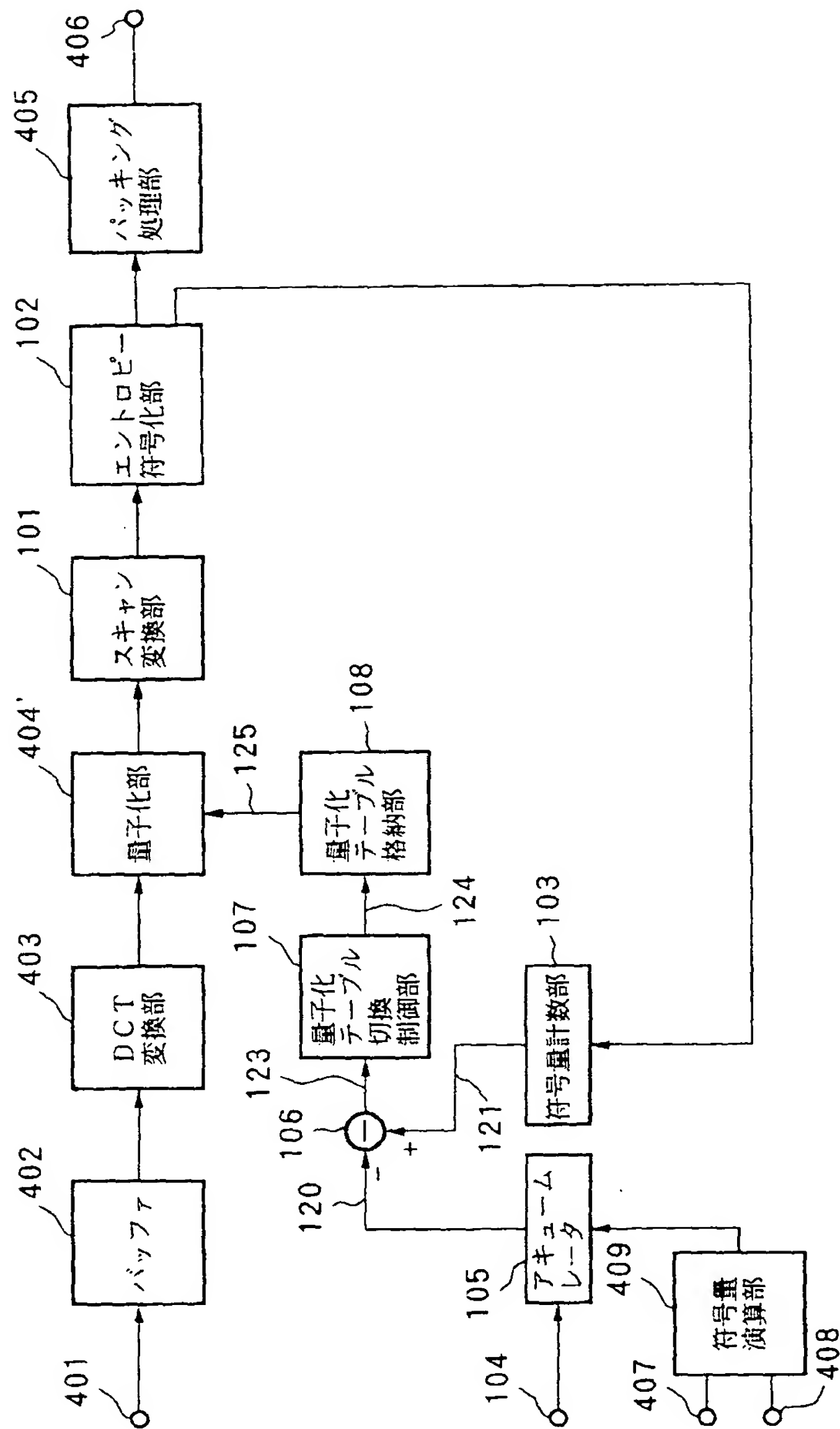
【図20】



【図11】

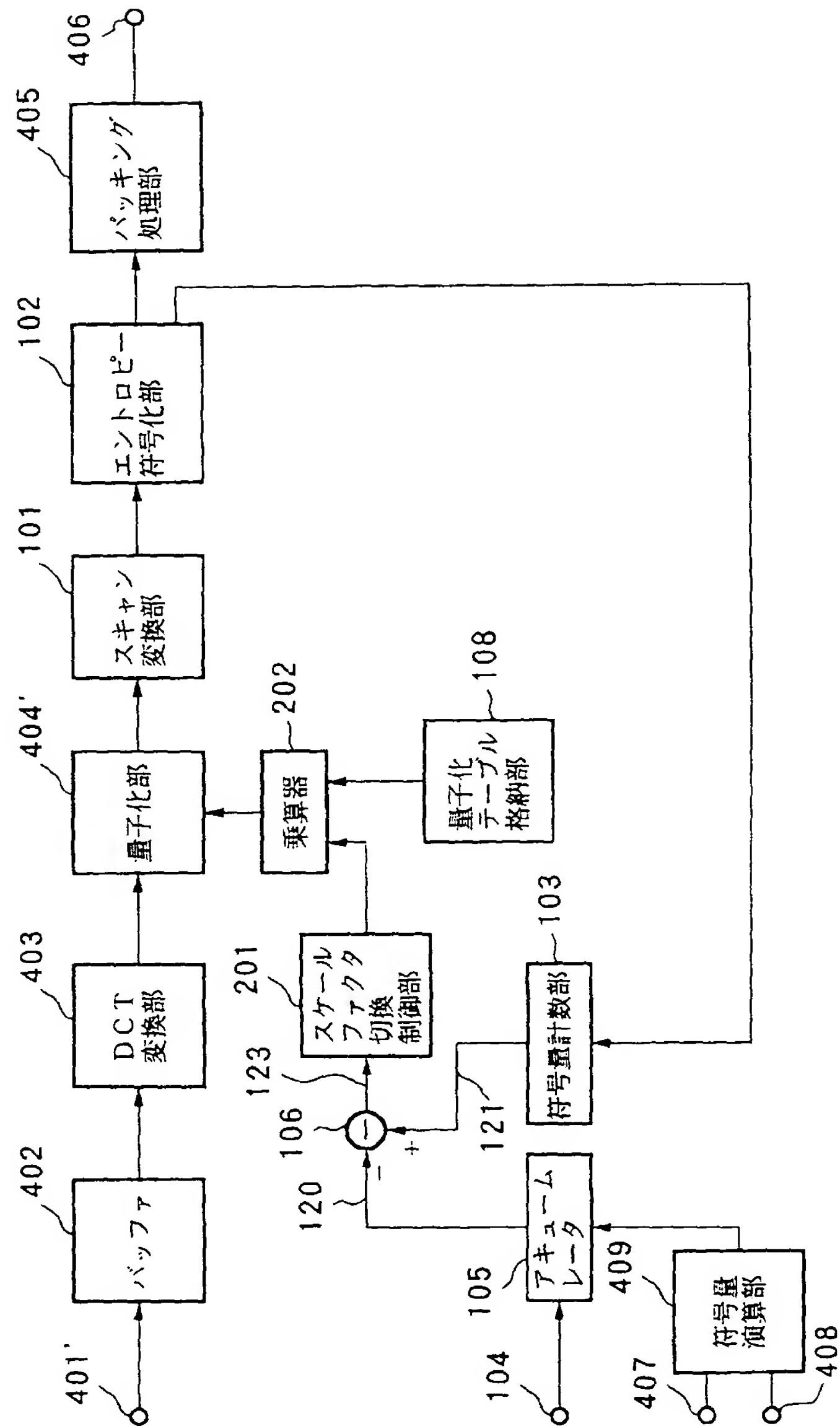


【図12】

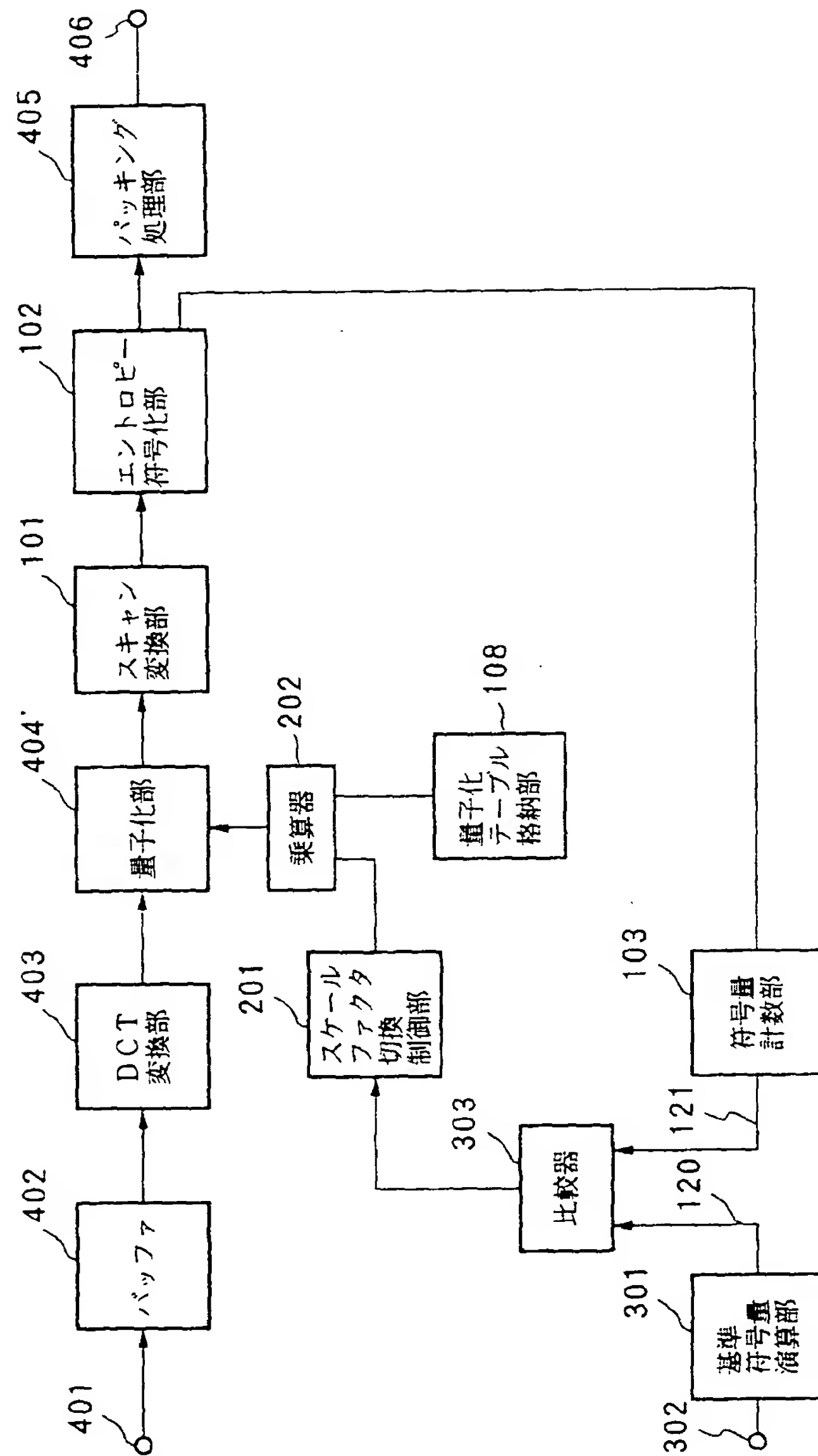




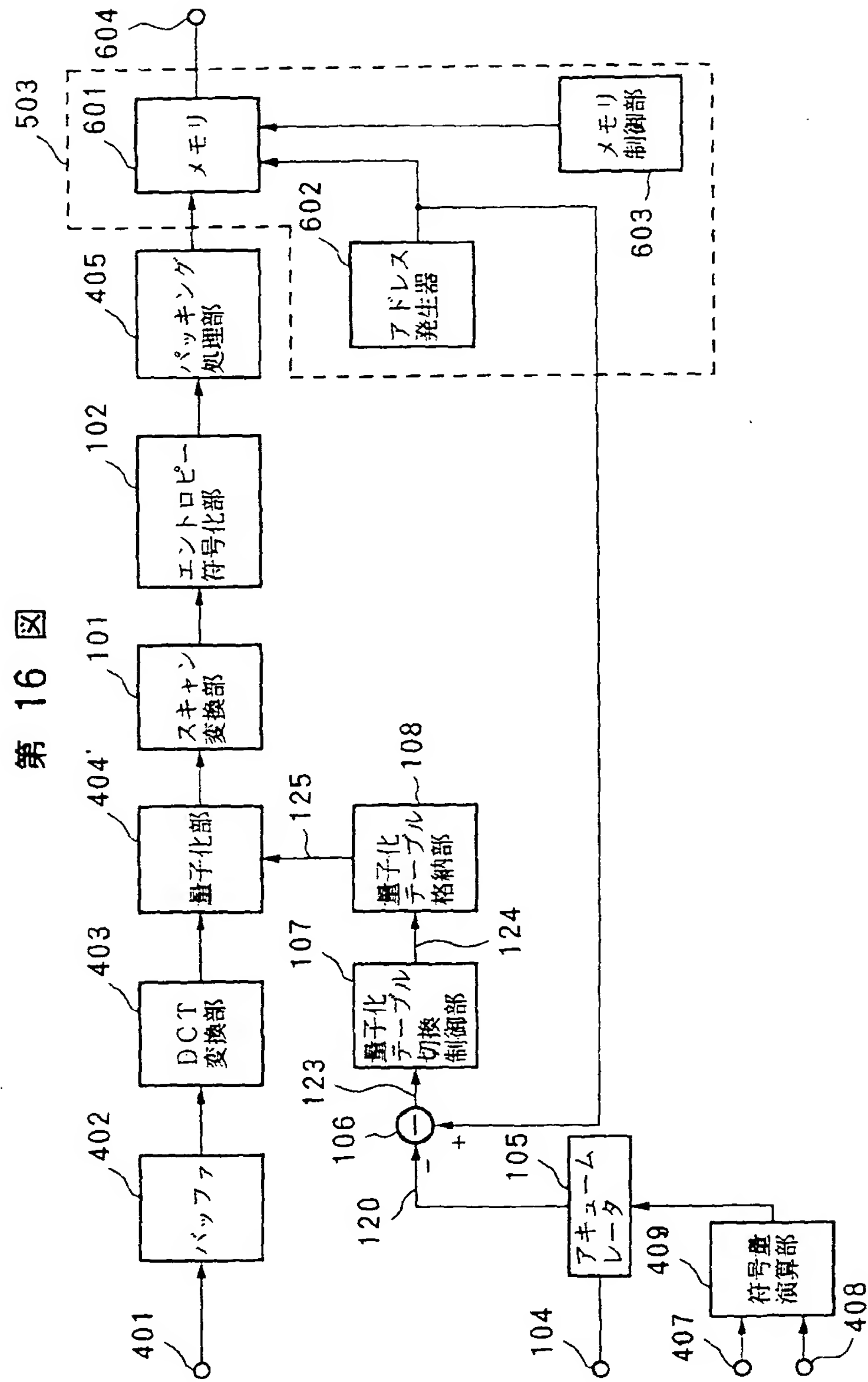
【図14】



【図15】

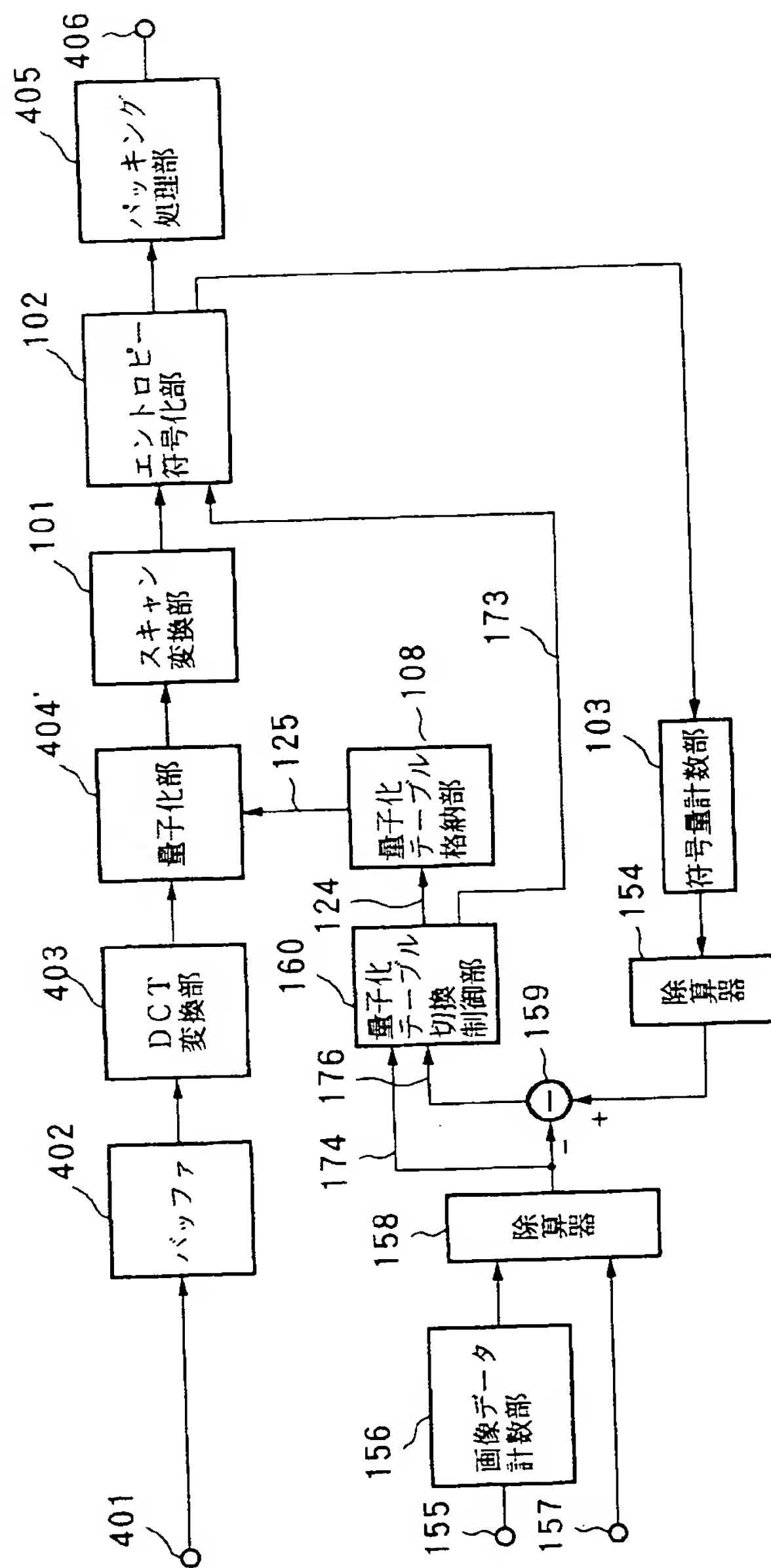


【図16】

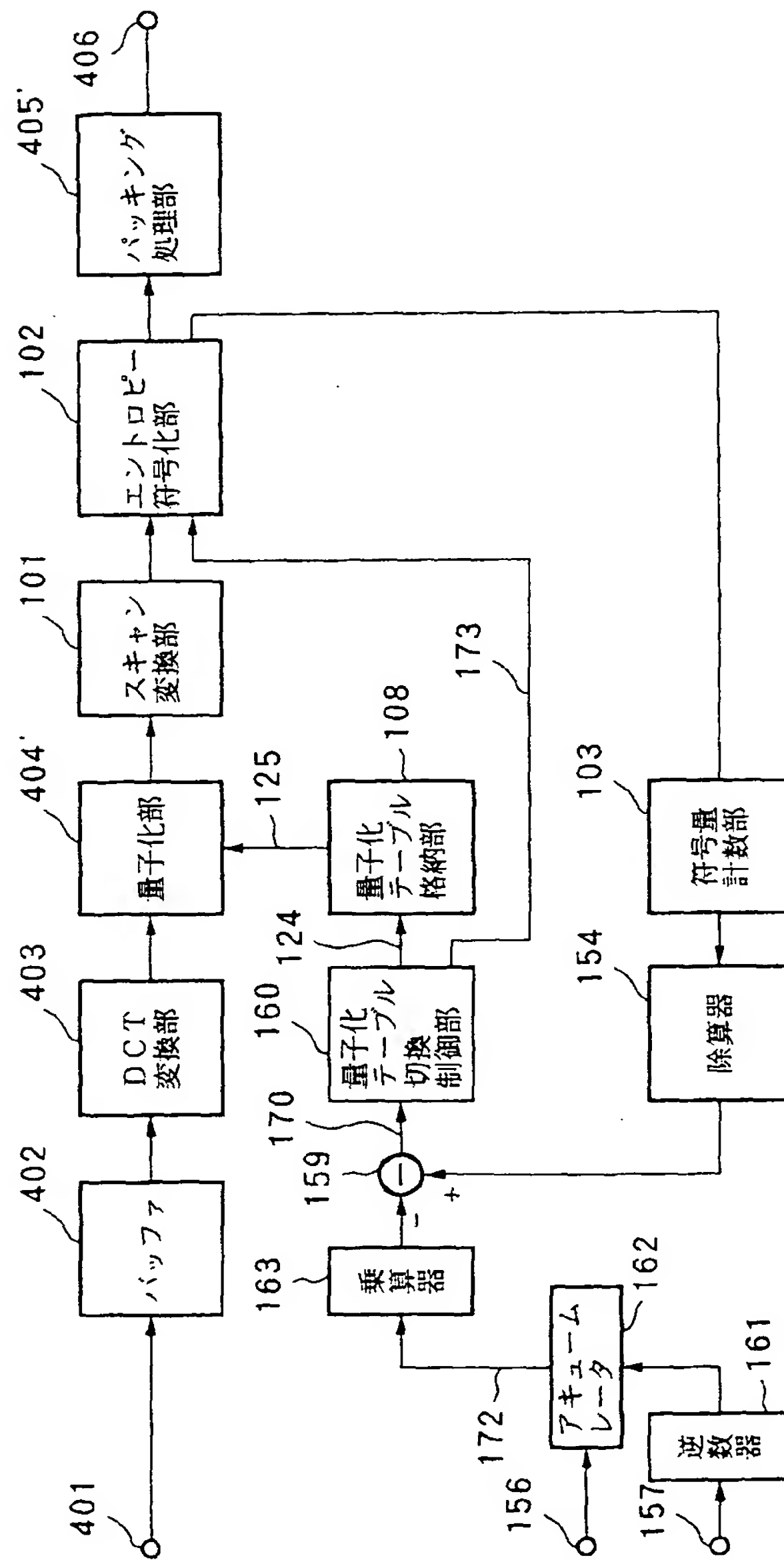




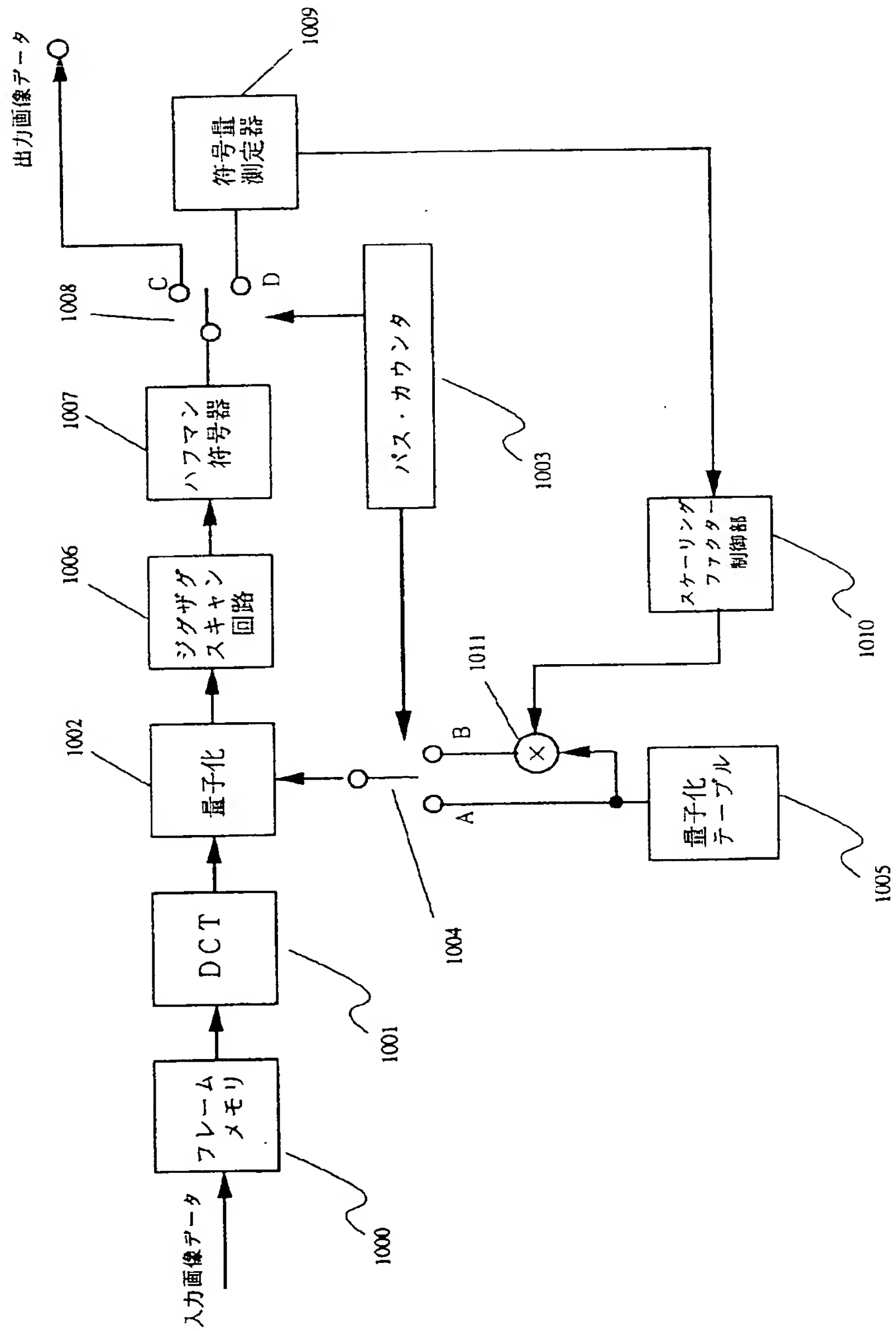
【図17】



【図18】



【図19】





フロントページの続き

(72)発明者 大木 丈二  
東京都大田区下丸子 3 丁目30番 2 号 キヤ  
ノン株式会社内